目　　录

1. 导论…………………………………………………………… 1
1.1. 要解决的问题 …………………………………………… 1
1.2. 数值近似解 ……………………………………………… 8
1.3. 例子——Euler 方法 …………………………………… 11
1.3.1. 误差估计……………………………………………… 15
1.3.2. 误差估计与实际误差的比较………………………… 17
1.3.3. 稳定性………………………………………………… 19
1.3.4. 舍入误差……………………………………………… 21
1.3.5. 由数值近似产生的扰动……………………………… 24
问题 ……………………………………………………… 27
2. 高阶单步方法………………………………………………… 30
2.1. Taylor 级数方法 ………………………………………… 30
2.2. Richardson 外插法 ($h=0$) ……………………………… 31
2.3. 二阶 Runge-Kutta 方法 ………………………………… 32
2.4. 显式 Runge-Kutta 方法 ………………………………… 37
2.4.1. 经典的 Runge-Kutta 方法…………………………… 42
2.4.2. Ralston Runge-Kutta 方法 ………………………… 43
2.4.3. Butcher 关于 Runge-Kutta 方法可达到阶的结果… 44
2.5. 隐式 Runge-Kutta 方法 ………………………………… 45
2.5.1. 隐式 Runge-Kutta 方法的实际应用 ……………… 48
2.6. 收敛性和稳定性 ………………………………………… 49
2.6.1. 显式 Runge-Kutta 方法的稳定区域 ……………… 50
2.6.2. 隐式 Runge-Kutta 方法的稳定区域 ……………… 52
问题 ……………………………………………………… 53
3. 方程组和高阶方程…………………………………………… 55

3.1. 单步方法应用于方程组 …… 56
3.2. 高阶方程简化为一阶方程组 …… 57
3.3. 高阶方程的直接方法 …… 58
3.3.1. Taylor 级数方法 …… 58
3.3.2. Runge-Kutta 方法 …… 59
问题 …… 62
4. 单步方法的收敛性、误差界和误差估计 …… 63
4.1. 向量和矩阵模 …… 64
4.2. 存在性和 Lipschitz 条件 …… 66
4.3. 收敛性和稳定性 …… 67
4.4. 误差界和收敛的阶 …… 72
4.5. 渐近误差的估计 …… 74
4.5.1. 由数值近似产生的扰动 …… 78
4.6. 误差界和估计定理的一般应用 …… 80
4.6.1. Taylor 级数方法 …… 81
4.6.2. Runge-Kutta 方法 …… 82
4.6.3. 对连续导数的要求 …… 83
4.7. 变步长 …… 83
问题 …… 85
5. 步长和阶的选取 …… 87
5.1. 阶的选取 …… 88
5.2. 步长的选取 …… 92
5.3. 误差的实际控制 …… 95
5.4. 局部截断误差的估计 …… 97
5.4.1. 步数加倍 …… 98
5.4.2. Runge-Kutta-Merson 方法 …… 102
问题 …… 103
6. 外插方法 …… 105

6.1. 多项式外插 ……………………………………105
6.1.1. 多项式外插的例 ……………………………………107
6.1.2. 舍入误差的影响 ……………………………………107
6.1.3. 稳定性 ……………………………………110
6.1.4. 高阶方法 ……………………………………110
6.2. 有理函数外插 ……………………………………112
问题 ……………………………………121
7. 多值或多步方法——导论 ……………………………………122
7.1. 多值方法 ……………………………………122
7.2. 显式多步方法——Adams-Bashforth 方法 ……………124
7.2.1. 系数的生成函数 ……………………………………129
7.2.2. 推导 Adams-Bashforth 方法的另外两个办法 ……131
7.2.3. Adams-Bashforth 方法的截断误差 ……………132
7.3. 隐式多步方法——Adams-Moulton 方法 ……………134
7.4. 预估-校正方法 ……………………………………137
问题 ……………………………………138
8. 一般的多步方法、阶和稳定性 ……………………………………140
8.1. 多步方法的阶 ……………………………………141
8.1.1. 给定 α, β 的一个确定另一个 ……………………144
8.1.2. 方法的主根 ……………………………………146
8.2. Milne 方法 ……………………………………147
8.2.1. 对于 $y' = \lambda y$ Milne 方法的稳定性 ……………149
8.3. 一般的多步方法的稳定性 ……………………………………151
8.3.1. 绝对稳定性 ……………………………………153
8.4. 四阶三步方法类 ……………………………………160
问题 ……………………………………163
9. 多值方法 ……………………………………165
9.1. 误差的性态 ……………………………………166
9.1.1. 预估-校正方法的稳定性 ……………………………167

9.2. 等价方法 ……………………………………………172
9.2.1. 影响表示式选取的因素…………………………173
9.2.2. Adams 方法的向后差分表示式 ………………178
9.2.3. Adams 方法的 Nordsieck 形式………………181
9.2.4. 改进的多步方法…………………………………183
9.2.5. 高阶方程…………………………………………185
9.3. 自动控制步长和阶 …………………………………189
问题 ……………………………………………………203
10. 多值方法的存在性、收敛性和误差估计……………204
10.1. 收敛性和稳定性 …………………………………207
10.1.1. 稳定性 ………………………………………210
10.1.2. 阶 ……………………………………………217
10.1.3. 相容性和收敛性 ……………………………225
10.2. 稳定多步方法的最高阶 …………………………233
10.3. 稳定多值方法的存在性 …………………………238
10.4. 标准形式多值方法阶的改进 ……………………241
10.5. 误差的渐近性质 …………………………………245
问题 ……………………………………………………249
11. 特殊问题的特殊方法 ………………………………251
11.1. Stiff 方程 …………………………………………251
11.1.1. 多步方法 ……………………………………255
11.1.2. A 稳定方法 …………………………………264
11.1.3. 基于 $\partial f/\partial y$ 的知识的方法 ……………………266
11.2. 代数方程和奇异方程 ……………………………268
11.3. 参数估计 …………………………………………273
问题 ……………………………………………………275
12. 方法的选取 …………………………………………277
12.1. 发展概貌 …………………………………………282
参考书目 ……………………………………………………284

1. 导　论

1.1. 要解决的问题

理论物理学家或化学家在用周围世界的模型进行研究工作上，花费了大量的时间．一个模型常常给成为一些变量的数学描述，其中有些变量是确实存在的，并且是可测的(如压力)，而另外一些变量可能仅仅是假定的（如一个新粒子的特征)．这些模型往往是常微分方程组，其中独立变量是时间，而因变量是物理变量．通过变量的代换，或者由于处在平衡状态的系统的解是与时间无关的，所以有时独立变量可以是一个物理变量．但是在本书中，我们将处处称时间为独立变量．下面要讨论的方法都不依赖时间的任何特殊性质，所以读者应准备用 x 来代替空间或者代替适合他所需要的量．

构造好模型，还要做两件事．首先，必须对它进行检验．这要求实验员在实验室做试验，将模型的运动状况与试验中观察到的结果相比较．其次根据对模型的运动进行的研究，理论工作者希望预估客观世界的变化．这些常微分方程除了很少情形能直接积分外，在要求得到一些数据的情况下，它们的积分要靠数值计算员来完成．假定刚好向我们提出了一对方程

$$\begin{cases} y' = (p - tq)y - z, \\ z' = y, \end{cases} \tag{1.1}$$

其中 y' 表示 dy/dt．在我们用计算机来计算之前，必须向物理学家提出哪些问题？而我们自己又一定要考虑哪些问题？

开始积分之前，我们必须得到为了确定一个问题的足够的信息．方程 (1.1) 含有二个未知常数 p 和 q，称之为参数，

而且必须知道它们的值．如果问题是要将试验结果与对 p 和 q 的各种值的数值积分结果进行比较，从根本上确定这些参数，可能要做大量的积分．也许需要研究更好的方法．这样一类问题放到倒数第二章来讨论；我们首先假定所有的参数均已给出．

考虑方程

$$\begin{cases} y' = z, \\ z' = -y, \end{cases} \tag{1.2}$$

它们有解:

$$\begin{cases} y = A\sin(t+\alpha), \\ z = A\cos(t+\alpha), \end{cases} \tag{1.3}$$

其中有两个积分常数 A 和 α．一般说来，N 个一阶方程的组有 N 个积分常数．如果我们要提供数值解，也就是对 t 的一些值计算 $y(t)$ 和 $z(t)$ 的值，那么，为了确定这些积分常数，必须给出足够的信息．如果因变量 y 和 z 的值给定在称为初值 t_0 的 t 的一个值上，一般来说，解就确定了(我们假定已作了一个变换，使 $t_0 = 0$．这不会影响问题或者所要讨论的方法)．确定 y 和 z 在未来时刻 t 上的值的问题叫作初值问题．另外，因变量的一些值可以给定在若干个不同的 t 值上，这叫作边值问题．这问题的最普通的形式是两点边值问题，其中函数值给在 t 的二个值上，比如 o 和 b．如果有 N 个一阶微分方程，且有 M 个因变量在 $t = 0$ 给定，则 $N-M$ 个值必须给定在 $t = b$ 上．在一些条件下，这将会得到一个解．

微分方程组的积分是一族曲线．例如，$y' = y$ 的积分为 $y = ce^t$，它是图 1.1 所表示的曲线族．选取初值是为了从族中选择一条曲线．如果积分常数多于一个，要画出解族就困难了．但是我们能够用考虑方程(1.2)的解族 (1.3) 来说明两

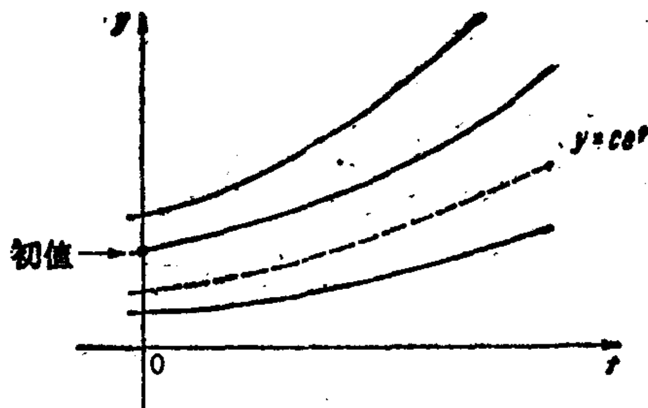


图 1.1. 一阶微分方程解族

点边值问题.如果 y 的值在 $t=0$ 给定，于是 z 在 $t=0$ 的不同的初值将给出图 1.2 所示的不同解组成的较小的族. 如果 y 在另外一个时刻 $t=b$ 给定,这个值足够用来选取所需要的族中的曲线. 在这个例子中，如果我们已选好 $y(0)=0$ 和 $b=\pi$，则对 z 的不同的初值,会得到图 1.3 所示的曲线族.在这种情形，指定 $y(\pi)$ 的值，不能得到完全确定的问题. 当 $y(\pi)=0$ 时,有无限多个解;而当 $y(\pi)\neq 0$ 时，没有解. 因此，我们看到边值问题不总是有唯一解的. 初值问题也是这样,但幸而还能够给出关于初值问题具有唯一解的适当准则. 由于处理两点边值问题的方法类型不同，我们只讨论关于初值问题的方法.

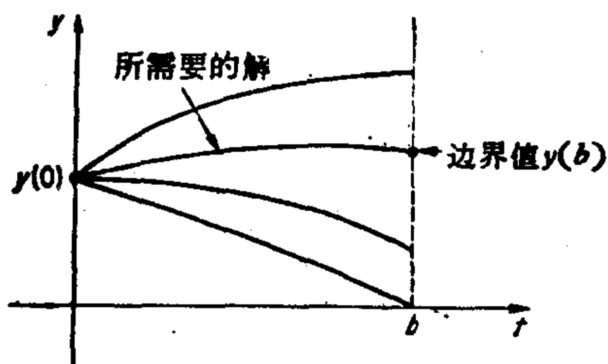


图 1.2. 两点边界值问题

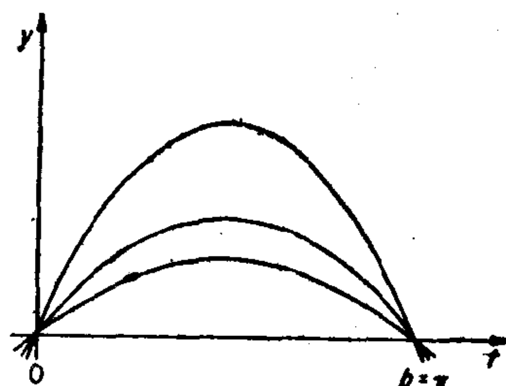


图 1.3. 没有唯一解的两点边值问题

有了确定所要求的解的充分材料之后，物理学家必须提出解的精度要求，用这个精度确定使用的方法，而且还要用它核对所提供的初值和参数的精度．虽然假定参数是准确的，但还要研究初值中的误差的影响．方程 $y'=y$ 的解族在图 1.1 中给出．由于初值误差使得选取了一个错误的解（假定因为初值误差得到的是虚线所表示的解而不是实线所表示的），则随着时间增加，曲线之间的差也增加．在这个例子中，以因子 e^t 的速度增加．我们称这种现象为方程的不稳定性．另一方面，若有方程 $y'=-y$，我们得到由图 1.4 所示的解族．在这种情形，当 t 增加时误差减小．这种现象称为方程的稳定性．如果初值给在 $t=0$ 上，积分积到 $t=b$，则最终的误差是初始误差的 e^{-b} 倍．对于微分方程 $y'=\lambda y$，其中 λ 是给定的，则误差增长 $e^{\lambda b}$ 倍．如果 $\lambda\leqslant 0$，初值误差没有增加，于是方程是稳定的．如果 $\lambda>0$，则方程不稳定．

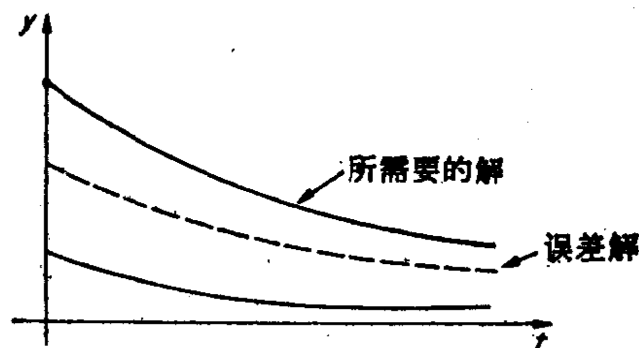


图 1.4．稳定的解族

显然，我们能够得到的精度部分地被初值的精度所限制．类似地，精度也将被参数中的误差所限制．一部分积分工作可以用来确定由这些误差所引起的解中的误差．假使我们能够直接积分方程，对所有可能的初值得到的解族进行考察，就能够做到这一点．但是，如果不能找到显式解，而用数值解，那么这个数值解将引进附加的误差．所能获得的最大精度将

被初值误差所限制.

虽然可以认为我们已经有求解方程的足够的信息，但是在开始数值积分之前，还必须考虑解的存在性. 许多数值方法均会得到一些数据，但是如果问题没有解，这些数据显然是毫无意义的. 特别，必须保证方程具有唯一的解. 一个通常的定理如下:

定理 1.1. 假设有微分方程 $y' = f(y, t)$，其中 $f(y, t)$在区域 $0 \leqslant t \leqslant b$ 中连续，并且存在常数L，使得

$$|f(y, t) - f(y^*, t)| \leqslant L\|y - y^*\|$$

对所有 $0 \leqslant t \leqslant b$ 和所有 y, y^* 均成立（这叫作 Lipschitz 条件，L 叫作 Lipschitz 常数），于是存在唯一的连续可微函数 $y(t)$，满足

$$y'(t) = f(y(t), t) \tag{1.4}$$

及初始条件 $y(0) = y_0$.

这个定理的证明可以在大多数关于常微分方程的书中找到，例如，见 Ince (1956)，第三章.

这里不要求 $f(y, t)$ 是可微的，但是，如果它是可微的，则 Lipschitz 条件保证有 $|\partial f/\partial y| \leqslant L$. 反过来，如果 f 对 y 是可微的，并且 $|\partial f/\partial y| \leqslant L$，则 f 满足 Lipschitz 条件. 这通常是验证条件是否满足的最容易的方法.

有些情形 Lipschitz 条件不满足，但是对解的附加的约束可以使它是唯一的. 例如，考虑方程

$$y' = \sqrt{1 - y^2} = f(y, t).$$

当 $y = 1$ 时，$\partial f/\partial y$ 是无界的. 事实上，它的解族为 $y = \sin(t + \alpha)$，如图 1.5 所示. $y = \pm 1$ 也是它的解，是一种特殊类型的解，它们构成解族的包络，即它们是处处与解族中至少一个解相切的曲线. 因此，图上由 $\sin(t)$ 的一部分、$y = 1$ 的一部分和 $\sin(t + \alpha_1)$ 的一部分组成的粗线对任何 α_1 也是

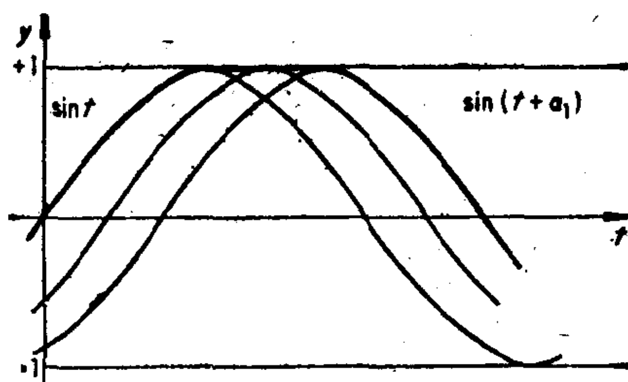


图 1.5. 具有包络线奇异性的解族

一个解，它具有连续的一阶导数．这样，对任意起始点 $-1\leqslant y_0\leqslant 1$，存在无穷多个解．但是，如果还要求具有连续的二阶导数，则存在唯一的解．在许多物理问题中，希望有若干阶的连续导数，这个事实可以保证有一个解，也可用来帮助选择方法．另外一个例子是方程

$$y'=\frac{y}{t}.$$

它既说明了这个问题，也说明初值问题可能没有解的事实，解族为 $y=ct$，如图 1.6 所示．由于它们都经过 $y=t=0$，初值不能给在奇异点 $t=0$ 上．

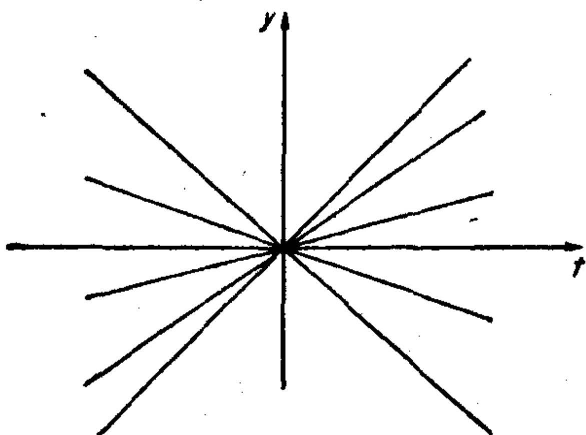


图 1.6. 具有一点奇异性解族

我们除了保证问题有解外，还必须保证它是适定的．所谓适定，是指在所述的问题中，微小的扰动只能引起解的微小的变化．这显然是一个有用的条件，因为对于解的数值近似，完全可能会引进扰动，使得正在求的是另外一个问题的解；把这些扰动控制得很小，使

得解能达到所需要的精确度是可以期望的. Lipschitz 条件是普通初值问题为适定的充分条件. 通过研究扰动问题

$$\begin{cases} z' = f(z, t) + \delta(t), \\ z(0) = y_0 + \varepsilon_0, \end{cases} \tag{1.5}$$

可以证实这一点，其中 $\delta(t)$ 和 ε_0 都是小的扰动. 令 ε 是由 $\max[|\varepsilon_0|, \max_{0\leqslant t\leqslant b}|\delta(t)|]$ 定义的模[1] $\|\varepsilon_0, \delta\|$. 如果 $\varepsilon(t)$ 为扰动后的解 z 和真实解 y 之间的差，用 (1.5) 减去 (1.4)，我们得到

$$\varepsilon'(t) = f(z,t) - f(y,t) + \delta(t), |\varepsilon(0)| = |\varepsilon_0| \leqslant \varepsilon.$$

于是

$$|\varepsilon'(t)| \leqslant |f(z,t) - f(y,t)| + |\delta(t)| \leqslant L|\varepsilon(t)| + \varepsilon.$$

经积分得

$$|\varepsilon(t)| \leqslant \frac{\varepsilon}{L}[(L+1)e^{Lt} - 1].$$

因此，扰动问题的解中最大改变量以

$$\max_{0\leqslant t\leqslant b}|\varepsilon(t)| \leqslant \|\varepsilon_0, \delta\| \cdot \frac{1}{L}[(L+1)e^{Lb} - 1] = k\varepsilon \tag{1.6}$$

为界，这里 k 与 ε 无关.

现在我们正式定义适定性.

定义 1.1. 常微分方程 (1.4) 对初始条件 y_0 是适定的，如果存在严格正的常数 k 和 $\tilde{\varepsilon}$，使得对于任意 $\varepsilon \leqslant \tilde{\varepsilon}$，只要 $|\varepsilon_0| < \varepsilon$ 并且对所有 $0 \leqslant t \leqslant b$ 有 $|\delta(t)| < \varepsilon$，则扰动问题 (1.5) 满足

$$|z(t) - y(t)| \leqslant k\varepsilon.$$

于是我们可以得到如下定理:

定理 1.2. 如果 $f(y, t)$ 满足 Lipschitz 条件，则 (1.4) 对任何初始条件都是适定的.

---

1) 模是一个正实数，它是一些量的大小的度量。后面对向量和矩阵我们将引进不同的模. 这儿的重要性质为由 $\|\varepsilon_0, \delta\| = 0$ 推出 $\varepsilon_0 = 0$ 和 $\delta(t) = 0$.

在许多问题中，我们不能对所有的 y，而只能在 y 空间的一个区域中得到 Lipschitz 条件(例如，如果 $f(y,t)=y^2$，$\partial f/\partial y$ 存在，并且在任何有限区域中有界)．当 y 属于这个区域时，定理 1.1 可以用来保证有唯一解．只要 z 和 y 都属于这个区域，定理 1.2 的证明就成立，所以，用 $\hat{\varepsilon}$ 来限制最大的扰动是必要的．例如，对于方程 $y'=\frac{1}{y^2}$，$y(0)>0$，只要扰动不使 y 小于 0，它的扰动都是有界的．因此，它对所有正的初始值 y_0 是适定的，但是，当 y_0 接近于零时，$\hat{\varepsilon}$ 要很小，而 k 要很大．

1.2. 数值近似解

得到微分方程的数值近似解有两个基本的途径．一个途径是把近似解表示成有限个独立函数之和，例如，截断的幂级数或者正交函数[1]展开式中的前面几项．这些方法通常比较适于手算，虽然将 Чебышев 多项式应用到常微分方程上来已经做了许多工作．

第二个途径是差分方法，这是我们在这本书中要研究的一种方法．解被它在一列离散点上的值来近似，这些离散点叫作节点．在我们讨论的大多数地方，将假定这些点是等距的，并且记作 $t_i=ih$，其中 h 是相邻两节点之间的距离．终点通常记为 $t_N=b$，因此有 $N=\frac{b}{h}$．然而将会看到，节距或步长会影响引进的误差，并且在区间的一部分上可能是一个好的步长，而在别处就不好了．因此，我们可以用可变的步长，在这种情形下，有

1) 一组函数 $\{\phi_n\}$ 称为对区间 $[a,b]$ 和权函数 $\omega(t)$ 正交，如果对 $m\neq n$ 有 $\int_a^b \omega(t)\phi_n(t)\phi_m(t)dt=0$．许多正交函数具有对数值近似目的有用的性质，见 A. Ralston (1965), p.93 的讨论．

$$t_{i+1} = t_i + h_i, \qquad t_0 = 0.$$

差分方法也叫作逐步方法，它提供了用 y 在 t_{i-1} 上或者前面的点上的值来计算第 i 步上关于 $y(t_i)$ 的近似值的规则. 我们称这个近似值为 y_i 在理想的情形，解能够用它在每个节点上的真实值来表示是理想的，所以，用节点之间的插值方法能够逼近到很高的精度. 但是，有两个问题妨碍这种理想的情形：首先，微分方程的精确解一般是不知道的，并且不能够计算，所以，求得的是另外一个能够计算的问题的解（这两个解之间的差称为截断误差）；其次，在其数值过程中，数字不能表示得完全精确（这些装置引进的变化称为舍入误差）. 因此，差分方法的解由有限个具有有限位正确的数字来表示，它含有两个误差源，舍入误差和截断误差. 差分方法也叫作离散变量方法，一般来说，比起级数展开方法，更适合于一般非线性问题的自动计算，并且是一般计算机子程序库中最常用的方法.

当我们想数值逼近一个解，自然关心能够使数值解对真实解达到多高的精度. 当我们选取一个方法时，它可能依赖于一个或许多个参数，例如，步长 h（当步长为可变时是 $\max(h_i)$）或者级数展开式中的项数. 我们想知道如何选取这些参数来达到需要的精度. 可能存在一个误差，比它小的误差是不能达到的. 在这一点上，我们粗略地定义收敛性概念如下：对于任何满足 Lipschitz 条件的问题，通过选取足够小的 h，可以达到任意的精确度. 当讨论特殊类型的方法时，这个定义将叙述得更加精确. 由于当 h 减小时，点数增加，因此，计算量也增加. 可以预料舍入误差的影响也要增加，因为舍入误差增多了. 于是，在定义收敛性时，必须要求方法中所要的计算都是精确完成的. 实际上，这表示当 h 减小时，在计算中要增加运算的位数.

前面我们确信问题为适定的，所以误差的影响是有界的。我们还需要知道，初值的微小变化在用这个方法得到的数值近似解中只产生有界的变化。我们称这个概念为稳定性。对于离散变量方法，我们不严格地定义成下面的意义：如果对于每个微分方程均存在 $h_0 > 0$，使得初值中小于一个固定量的变化在所有 $0 < h \leqslant h_0$ 的数值解中产生有界的变化，于是方法是稳定的。当讨论特殊类型的方法时，这个定义显得更加精确。我们看到，象适定性是对问题来说的一样，稳定性是关于方法的。注意，稳定性不要求收敛性，然而反过来，是真确的。因此，"方法" $y_n = y_{n-1}$，$n = 1, 2, \cdots, N$ 是稳定的，但是，除平凡问题 $y' = 0$ 外，对任何问题都不收敛。

稳定性和收敛性概念都涉及 $h \to 0$ 时的极限过程。实际上，必须用有限步来计算并且只关心对于非零 h 产生的误差的大小。特别，我们要知道每一步所引进的误差（截断与舍入）在结果上产生或大或小的影响。因此，我们如下定义绝对稳定性："对于给定的步长和给定的微分方程，如果由一个节点值 y_n 上大小为 δ 的扰动所引起的其后值 $y_m (m > n)$ 上的变化都不大于 δ，则方法是绝对稳定的。"可惜，这个定义太依赖于问题本身，所以，我们利用"试验方程"的思想。我们将对微分方程 $y' = \lambda y$ 定义绝对稳定，其中 λ 为复常数[1]，并且称绝对稳定区域为 h（非负实数）和 λ 的值的集合，对于这些值，单个

---

1) 对于线性方程组 $\mathbf{y}' = A\mathbf{y}$，如果 A 可对角线化，则可以归结成一组试验方程。如果 $SAS^{-1} = \Lambda$ 是将 A 变换成具有对角线元素 λ_i 的对角线矩阵 Λ 的相似变换，则可以写 $\mathbf{z} = S\mathbf{y}$，并且得到

$$S^{-1}\mathbf{z}' = AS^{-1}\mathbf{z}$$

或

$$\mathbf{z}' = SAS^{-1}\mathbf{z} = \Lambda\mathbf{z}.$$

这是一组独立的方程，形式均为 $z_i' = \lambda_i z_i$。对于一般的非线性方程，可以将 λ_i 看成 Jacobi 矩阵 $\partial\mathbf{y}'/\partial\mathbf{y}$ 的特征值。这些值确定了系统的一阶近似的局部性质。这些特征值 λ_i 可以是复的。

y_n 上量的扰动将在以后的值上产生一步一步不增加的变化.

1.3. 例子——Euler 方法

考察最简单的方法——Euler 方法是有益的，因为它容易分析，并且常常作为构造存在性证明的基础. 在 Euler 方法中，因变量在一点的值是从前一点由线性外插来计算的. 我们考虑单个方程

$$y' = f(y, t)$$

并且假定$y(0) = y_0$已给定. 于是可以计算 $y_0' = f(y_0, 0)$. 由这个值用 Taylor 级数的前两项可以计算 $y(h)$ 的近似值

$$y(h) \cong y_0 + hy'(0).$$

令 $t_1 = h$，并记 $y(t_1)$ 的近似值为 y_1，于是有

$$y_1 = y_0 + hf(y_0, t_0).$$

一般，由当时的值用公式

$$y_{n+1} = y_n + hf(y_n, t_n) \tag{1.7}$$

确定下一点的值，其中

$$t_n = nh.$$

从图 (1.7) 看出，通常每一步都是从解族的一条跑到另一条上去，所以，我们料想到解的精确度与方程的稳定性是紧密相关的. 如果方程是非常稳定的，则前面几步中的误差只有很小的影响. 另一方面，如果它们是不稳定的，则前面的误差有较大的影响.

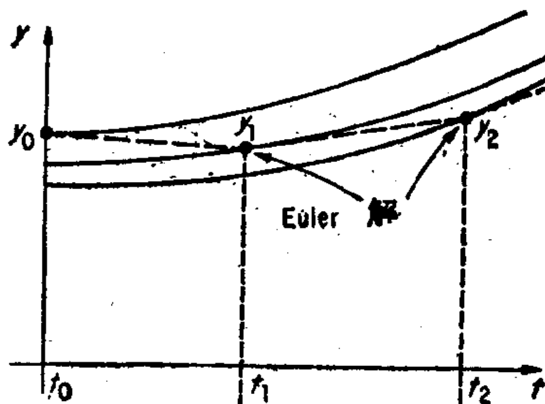


图 1.7. Euler 方法图示

假使对于给定的步长 h，Euler 方法的误差太大，怎么办?

能够使它减小吗？换句话说，当 $h \to 0$ 时这个方法收敛吗？答案在下面的定理中给出.

定理 1.3. 如果对于 $0 \leqslant t \leqslant b$ 和所有的 y，$f(y, t)$ 对 y 满足 Lipschitz 条件并且对 t 是连续的，$h = \frac{t}{n}$，序列 $y_i(i = 1, \cdots, n)$ 由 (1.7) 确定并且 $y_0 \to y(0)$，则当 $n \to \infty$ 时对 t 一致地有 $y_n \to y(t)$，其中 $y(t)$ 为方程 (1.4) 的初值为 $y(0)$ 的解.

我们称 y_0 为起始值，以示与初值 $y(0)$ 的区别. 事实上，我们只能希望在数值计算中的起始值当 h 减小和使用更高的精度时逼近初值. 在这个定理中，假定 (1.7) 是在无舍入误差时求解的.

虽然我们要求对于一切 y 都有 Lipschitz 条件，但是，实际上只需要它在一个包含解 $y(t)$ 在其内部的闭域 R 中成立. 步长可以选得足够小，使得数值解也保持在 R 内，条件就适合了. 在很多定理中，为了从用到的各种连续性推出 f 及其导数有界，数值解和 $y(t)$ 都属于一个闭域的事实是需要的. 定理 1.3 的证明将在下面讨论. 它归结成推导误差

$$e_n \coloneqq y_n - y(t_n)$$

的界，并证明这个界可以变得任意小. 对函数 f 和解 y 作些附加的假设，可以得到更好的误差界. 如果界只依赖于方程本身，而不依赖于关于解 $y(t)$ 的知识，称它为先验界. 另外，若它需要关于解的性质的知识，这种误差界称为后天界. 误差界通常比数值积分中产生的实际误差大得多，所以，我们有时将误差估计做成渐近形式，即找一个函数 $e(t)$，使当 $h \to 0$ 时有

$$\frac{e_n}{e(t_n)} = 1 + O(h)^{1)}.$$

---

1) 用 $O(h)$. 表示 h 的任何函数，使得存在不依赖于 h 的常数 h_0 和 k，对所有 $|h| \leqslant h_0$ 有 $|O(h)| \leqslant kh$.

误差估计通常是后天界的形式。

定理 1.3 的证明:

为了得到先验误差界，我们写出

$$y(t_{n+1}) = y(t_n) + hf(y(t_n), t_n) - d_n, \tag{1.8}$$

d_n 称为局部截断误差. 它是解不满足差分方法的那一部分量. 从 (1.7) 减去此式，得到

$$e_{n+1} = e_n + h(f(y_n, t_n) - f(y(t_n), t_n) + d_n; \tag{1.9}$$

记成

$$f(y_n, t_n) - f(y(t_n), t_n) = e_n L_n, \tag{1.10}$$

于是

$$e_{n+1} = e_n(1 + hL_n) + d_n.$$

这是关于 e_n 的差分方程[1]. 误差 e_0 是已知的，如果我们又知道 L_n 和 d_n，就能求解. 对 $|L_n|$ 有一个界 Lipschitz 常数 L，假定还有 $D \geqslant |d_n|$，于是有

$$|e_{n+1}| \leqslant |e_n|(1 + hL) + D. \tag{1.11}$$

这个差分方程在本书中经常出现，并导出下面的引理.

引理 1.1. 如果 $|e_n|$ 满足 (1.11)，并且 $0 \leqslant nh \leqslant b$，则

$$|e_n| \leqslant D\frac{(1 + hL)^n - 1}{hL} + (1 + hL)^n |e_0|$$

$$\leqslant \frac{D}{hL}(e^{Lb} - 1) + e^{Lb}|e_0|. \tag{1.12}$$

(1.12) 中第一个不等式由归纳法得出. 对于 $n = 0$，无需证明. 假定它对于 n 是正确的，从 (1.11) 有

$$|e_{n+1}| \leqslant D\frac{(1 + hL)^n - 1}{hL}(1 + hL) + D + (1 + hL)^{n+1}|e_0|$$

1) 很多书中都讨论了差分方程，象 P. Henrici (1963) 的第三章. 这是特别简单的差分方程，更复杂的情形在后面几章讨论.

$$= D\frac{(1+hL)^{n+1}-(1+hL)+hL}{hL}+(1+hL)^{n+1}|e_0|$$

$$= D\frac{(1+hL)^{n+1}-1}{hL}+(1+hL)^{n+1}|e_0|.$$

因此，(1.12) 对于 $n+1$ 成立．由于 $nh \leqslant b$，并且对于 $hL \geqslant 0$，有 $1+hL \leqslant e^{Lh}$，所以 $(1+hL)^n \leqslant e^{Lnh} \leqslant e^{Lb}$．由此事实得到 (1.12) 中第二个不等式．引理证毕．

为了继续证明定理 1.3，需要研究局部截断误差的界 D．由 (1.8)，

$$-d_n = y(t_{n+1}) - y(t_n) - hf(y(t_n), t_n).$$

由中值定理，对 $0 \leqslant \theta \leqslant 1$，我们得到

$$\begin{aligned}|d_n| &= |hf(y(t_n+\theta h), t_n+\theta h) - hf(y(t_n), t_n)| \\ &\leqslant h|f(y(t_n), t_n+\theta h) - f(y(t_n), t_n)| \\ &\quad + h|f(y(t_n+\theta h), t_n+\theta h) - f(y(t_n), t_n \\ &\quad + \theta h)| \leqslant h|f(y(t_n), t_n+\theta h) - f(y(t_n), t_n)| \\ &\quad + hL|(y(t_n+\theta h) - y(t_n))|. \end{aligned} \tag{1.13}$$

最后一项可以利用中值定理来处理，得到一个界

$$L\theta h^2|y'(\xi)| \leqslant h^2 LZ,$$

其中 $Z = \max|y'(t)|$，由于 y 和 f 在闭区域中连续，它是存在的．(1.13) 中第一项的处理依赖于假定．不加另外的假定证明就能进行下去，例如见 Henrici (1962) 的第 1.2.4 节．但是，如果假定 $f(y, t)$ 对 t 也满足 Lipschitz 条件（在实际情形，至少能够分段满足），我们能使 (1.13) 的第一项以 $K\theta h^2$ 为界，这里 K 是将 f 看成 t 的函数时的 Lipschitz 常数．因此，

$$|d_n| \leqslant h^2(K + LZ) = D.$$

所以，从 (1.12) 得到

$$|e_n| \leqslant h\frac{K+LZ}{L}(e^{Lb}-1) + e^{Lb}|e_0|. \tag{1.14}$$

这样，当 $h \to 0$ 时，如果 $|e_0| \to 0$，则数值解收敛．

1.3.1. 误差估计

在上面证明收敛性时，我们推导了对于解的误差的界(1.14)．这表明：如果 $\partial f/\partial t$ 存在、连续且有界，则误差按 $O(h)$ 变化.一般来说函数 f 是可微的，并且能够计算界 K, L 和 Z，至少对于 y 空间的一个有限区域，可证明真解与数值解属于其内部．但是，这样得到的误差的界可能不会很好，因为必须选取 $|y'|$, $|\partial f/\partial y|$ 和 $|\partial f/\partial t|$ 的最大值．假使我们有解的一些知识，而且假定它的二阶导数连续且小于一个已知数，例如 C，就能得到一个较好的界．首先，在 t_n 用带余项的 Taylor 级数来表示d_n，对 $\xi\in(t_n, t_{n+1})$ 得到

$$-d_n = y(t_{n+1}) - y(t_n) - hf(y(t_n), t_n) = \frac{h^2}{2}y''(\xi).$$

因此有

$$|d_n| \leqslant \frac{Ch^2}{2}$$

和

$$|e_n| \leqslant \frac{h}{2}\frac{C}{L}(e^{Lb}-1) + e^{Lb}|e_0|. \tag{1.15}$$

这是一个后天的界，因为它依赖于解的二阶导数．但是，当 y'' 存在并且连续时，将 y'' 写成

$$y'' = (y')' = \frac{df}{dt} = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial y}y',$$

我们就能将其转换成先验界．

要改进由(1.15)给出的误差的界是困难的，但是，我们能够另外寻找一个误差估计．因为我们知道，如果 y'' 存在，误差按 $O(h)$ 变化，我们想将其表达成 $e_n = h\delta_n$ 的形式，并且得到 δ_n 的大小的估计．为此，我们假定 y 具有连续的三阶导数，于是，我们能写出

$$d_n = -\frac{1}{2}h^2y''(t_n) - \frac{h^3}{6}y'''(\xi),$$

其中 $\xi \in (t_n, t_{n+1})$ 而且方程 (1.9) 可以写成

$$e_{n+1} = e_n + h(f(y_n, t_n) - f(y(t_n), t_n)) - \frac{1}{2}h^2y''(t_n) - \frac{h^3}{6}y'''(\xi). \qquad (1.16)$$

由具有余项的 Taylor 级数，括弧中的项可以表成

$$e_n\frac{\partial f}{\partial y}(y(t_n), t_n) + \frac{1}{2}e_n^2\frac{\partial^2 f}{\partial y^2}(\xi, t_n).$$

如果 (1.16) 中的 e_n 用 $h\delta_n$ 代替，我们得到

$$\delta_{n+1} = \delta_n + h\left(\delta_n\frac{\partial f}{\partial y} - \frac{1}{2}y''\right) + \frac{1}{2}e_n^2\frac{\partial^2 f}{\partial y^2}(\xi) - \frac{h^2}{6}y'''(\xi). \qquad (1.17)$$

所有导数除了指出者外均在 t_n, $y(t_n)$ 上求值. 从 (1.15) 有 $|e_n| < hK_1$, 因此 (1.17) 给出

$$\delta_{n+1} = \delta_n + h\left(\delta_n\frac{\partial f}{\partial y} - \frac{y''}{2} + hc_n\right), \qquad (1.18)$$

其中

$$|c_n| \leqslant \frac{K_1^2}{2}\max\left|\frac{\partial^2 f}{\partial y^2}\right| + \frac{1}{6}\max|y'''| \leqslant K_2.$$

注意,(1.18) 为微分方程

$$\frac{d\delta}{dt} = g(t)\delta - \frac{1}{2}y'' + hc(t), \quad \delta(0) = \frac{e_0}{h} \qquad (1.19)$$

用 Euler 方法得到的数值解,其中 $c(t)$ 为当 $t = t_n$ 时取 c_n 值的任意函数,而 $g(t) = \partial f/\partial y$ 在 $y = y(t)$ 上求值. 象(1.14)所表明的, (1.19) 的实际解等于数值解 (1.18) 加上 $O(h)$;又由于问题是适定的(见定理 1.2), 因此, (1.19) 的实际解为方程

$$\frac{d\delta}{dt} = g(t)\delta - \frac{1}{2}y'', \quad \delta(0) = \frac{e_0}{h} \qquad (1.20)$$

的解加上 $h \max c(t) \leqslant KK_2$ 的有界倍. 这样, 我们已证明了

下面的误差估计.

定理 1.4. 如果 y 三次连续可微,则 Euler 方法中的误差形如

$$e_n = h\delta(t_n) + O(h^2), \tag{1.21}$$

其中 $\delta(t)$ 为 (1.20) 的解.

1.3.2. 误差估计与实际误差的比较

我们已经得到误差的界 (1.14) 和 (1.15) 以及误差的估计 (1.21),现在在几个例子中将它们与实际误差进行比较.考虑 $y' = y$, $y(0) = 1$, 解为 $y(t) = e^t$, 所以 $y(1) \cong 2.71828$. Euler 方法给出 $y_{n+1} = y_n + hy'_n = y_n(1 + h)$, 因此, $y_N = (1 + h)^N$. 如果 $t_N = 1$, $y_N = (1 + h)^{\frac{1}{h}}$, 当 $e_0 = 0$ 时,误差界(1.14) 给出

$$|e_N| \leqslant h2.71828(2.71828 - 1) = 4.67077h,$$

而 (1.15)给出

$$|e_N| \leqslant h1.35914(2.71828 - 1) = 2.33539h,$$

误差估计 (1.21) 给出

$$e_N \cong h\delta(t),$$

其中 δ 是方程

$$\delta'(t) = \delta(t) - \frac{1}{2}e^t, \delta(0) = 0$$

的解. 因此, $\delta(t) = -\frac{1}{2}te^t$, $\delta(1) = -1.35914$. 对 h 的若干个值, 结果列在表 1.1 中.

我们看到, 当 $h \to 0$ 时, 误差接近估计值,但误差的界十分不理想. 现在考察

$$y' = -y, \quad y(0) = 1,$$

解为 $y = e^{-t}$, 所以, 在 [0.1] 中 $|y'|$ 和 $|y''|$ 的最大值是 1. 因此,界 (1.14) 和 (1.15) 分别给出 1.71828h 和 0.85914h.

表 1.1. 对 $y' = y$ 的实际误差及界和估计

h	y_N	e_N	e_N/h	界 (1.14)/h	界 (1.15)/h	估计 (1.21)/h
1	2	−0.71828	−0.71828	4.67077	2.33539	−1.35914
$\frac{1}{2}$	2.25	−0.46828	−0.93656	4.67077	2.33539	−1.35914
$\frac{1}{4}$	2.44141	−0.27688	−1.10750	4.67077	2.33539	−1.35914
$\frac{1}{8}$	2.56578	−0.15250	−1.21998	4.67077	2.33539	−1.35914
$\frac{1}{16}$	2.63793	−0.08035	−1.28565	4.67077	2.33539	−1.35914
$\frac{1}{32}$	2.67699	−0.04129	−1.32134	4.67077	2.33539	−1.35914
$\frac{1}{64}$	2.69734	−0.02094	−1.33997	4.67077	2.33539	−1.35914

由于 $y'' = e^{-t}$, δ 由

$$\delta'(t) = -\delta(t) - \frac{1}{2} e^{-t}, \delta(0) = 0$$

给出，所以 $\delta(t) = -\frac{1}{2} te^{-t}$ 和 $\delta(1) = -0.18394$. 对于各种步长误差的比较，由表 1.2 给出. 从表看出，由方程 (1.21) 得

表 1.2. 对 $y' = -y$ 的实际误差及界和估计

h	y_N	e_N	e_N/h	界 (1.14)/h	界 (1.15)/h	估计 (1.21)/h
1	0	−0.36788	−0.36788	1.71828	0.85914	−0.18394
$\frac{1}{2}$	0.25	−0.11788	−0.23576	1.71828	0.85914	−0.18394
$\frac{1}{4}$	0.31641	−0.05147	−0.20589	1.71828	0.85914	−0.18394
$\frac{1}{8}$	0.34361	−0.02427	−0.19416	1.71828	0.85914	−0.18394
$\frac{1}{16}$	0.35607	−0.01181	−0.18889	1.71828	0.85914	−0.18394
$\frac{1}{32}$	0.36206	−0.00582	−0.18637	1.71828	0.85914	−0.18394
$\frac{1}{64}$	0.36499	−0.00289	−0.18515	1.71828	0.85914	−0.18394

到的估计给出比实际误差较小的数，但是，当步长减小时，实际误差接近于估计误差.

由方法的单独一步引进的附加误差通常称作局部截断误差. 它是方法给出的值与微分方程的解之间的差，而这个解通过这一步开始时的值. 在 Euler 方法中，将

$$y(t_{n+1}) = y(t_n) + hy'(t_n) + \frac{1}{2} h^2 y''(\xi)$$

与 Euler 公式比较，我们能够找到它．局部截断误差是附加项 $\frac{1}{2}h^2y''(\xi)$．注意，除了因子 h^2 外，这就是误差估计方程 (1.20) 中的非齐次项．在 Euler 方法一步中的截断误差与初值误差一样影响下一步，如果它将数值解移到解族的同样的新的解上的话．后面几章研究一种方法，在这种方法中，一步所引进的误差可能会影响后面若干步，因为为了得到一个新的 t 上较好的近似值，用到了若干个不同的 t 上的信息．在这种情形，局部误差的影响按类似的方式加到与方程 (1.20) 等价的方程中去．

1.3.3. 稳定性

一个计算值从 y_n 到 z_n 的变化，使我们求解

$$z_{m+1} = z_m + hf(z_m, t_m)$$

来代替解 (1.7)．从此式减去 (1.7)，并令 $e_n = z_n - y_n$，得到

$$|e_{m+1}| \leqslant |e_m| + hL|e_m|$$

或

$$|e_N| \leqslant (1 + hL)^{N-n}|e_n| \leqslant e^{bL}|e_n|.$$

它是引进的误差 $|e_n|$ 的有界倍，并且不依赖于 h．因此，方法是稳定的．

为了考察绝对稳定性，我们研究方程 $y' = \lambda y$．对此方程有

$$y_{n+1} = y_n + \lambda h y_n = (1 + \lambda h)y_n.$$

因此，在区域 $|1 + \lambda h| \leqslant 1$ 中，Euler 方法是绝对稳定的，这个区域是复 λh-平面上中心在 $(-1, 0)$ 的单位圆．由下面的例子：

$$y' = -1000(y - t)^2 + 2t,$$

$$y(0) = 0, \text{计算 } y(1).$$

我们可以清楚地看到绝对稳定性的作用．在机器上以 10 位有

效数字的精度用步长为 $10^{-m}(m=0, 1, 2, \cdots, m)$ 的 Euler 方法解这个问题，结果由表 1.3 列出.

表 1.3. 绝对稳定性的效果

h	N	$y(1)$
1	1	0
0.1	10	$0.90438207503\times10^{16}$
0.01	100	溢出
0.001	1000	0.99999900001
0.0001	10000	0.99999990000
0.00001	100000	0.99999998997

溢出表示超过了机器所能容纳的最大数，在现在的情形为 10^{38}. 发生这种现象是因为对于这个问题 $\partial f/\partial y$ 为 -1000，所以当 $h=0.01$ 时，误差每步扩大 $\left(1-\frac{1000}{100}\right)=-9$ 倍. 在 100 步中，第一步的误差在数量级上差不多增加 10^{100} 倍. 只要 $|1+\lambda h|$ 小于 1，结果就合理，并且收敛于正确解. 步长为 10^{-k} 的误差非常接近于 10^{-k-3}，这表示在绝对稳定性区域内，误差线性依赖于 h.

不考虑绝对稳定性，误差界 (1.14) 和 (1.15) 是成立的，但是对于这个问题，$L=1000$ 和 $C=2$，因而 (1.15) 仅得出界

$$|e_n|\leqslant\frac{h}{1000}(e^{1000}-1).$$

对实际值来说，它太大了. 对这个特殊情形，我们能够指出，如果 $e_n=y_n-t_n^2$，则

$$e_{n+1}=e_n-1000he_n-h^2.$$

因此

$$e_N=\frac{h}{1000}((1-1000h)^N-1).$$

对于 $h \leqslant 0.001, N = h^{-1}$,

$$e_N \cong \frac{h}{1000}.$$

但是,对于 $h \geqslant 0.01, N = h^{-1}$,

$$e_N \cong \frac{(1000)^{N-1}}{N^{N+1}}.$$

这表明严格依赖于绝对稳定性.

1.3.4. 舍入误差

我们假定了在 Euler 方法中用到的数值计算都是精确完成的. 事实上,由于计算中用了有限位数字,总要产生舍入误差. Henrici (1962) 非常详细地考察了定点机上的舍入误差,读者可以参看那里的统计处理. 我们粗略地考察浮点舍入误差的影响. 用 Euler 公式计算的每一步,由于数值不精确产生了附加项 r_n, 因此,有

$$y_{n+1} = y_n + hf(y_n, t_n) + r_n, \tag{1.22}$$

并且 r_n 象附加的局部截断误差那样起作用. 如图 1.7 所示,每一步它都可以将解转移到族中另外一个解上. 如果存在一个函数 $r(t)$, 满足 $r(t_n) = r_n$, 它就能够与截断误差一起在方程 (1.20) 中表示出来. 假定存在那样的函数,于是,导出 (1.20) 的分析将给出

$$e_n = h\delta(t_n) + O(h^2),$$

其中 $\delta(t)$ 是方程

$$\delta' = g(t)\delta - \frac{1}{2}y'' + h^{-2}r(t) \tag{1.23}$$

的解. 与 h^2 从局部截断误差 $\frac{1}{2}h^2y''$ 中去掉一样, 在 (1.23) 中出现 h^{-2} 项. 这向我们表明 $e_n = O(h^{-1}|r| + h)$,其中 $|r|$ 为 $r(t)$ 依赖于 h 的一种度量.

如果 $|r|$ 与 h 无关，当位数保持固定时就是这样，则开始当 h 减小时，截断误差减小，总的误差也减小，然后由于舍入误差愈来愈大，总的误差增加．这由图 1.8 表出．

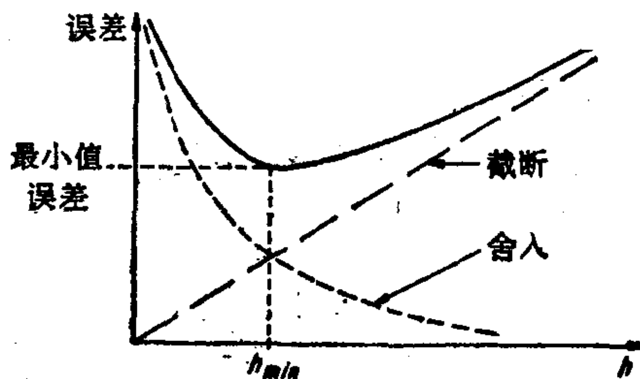


图 1.8．误差化为力的函数

在数字机上进行的大多数积分中，不需要达到最小的误差，可以用一个适当的超过 $h_{\min}$ 的步长．但是，如果要求更高的精度，h 必须小于 $h_{\min}$． 为了减小 $h^{-1}|r|$，必须增加数字的位数．

如果最坏的舍入误差 ε 在每一步都出现，则 $r(t)=\varepsilon$ 和 $e_n=O(h^{-1})$．换句话说，在这种情形，当 $h\to 0$ 时数值解不收敛．随便一看可能认为是对的．当每步引进的误差量固定，随着运算次数的增加，引进的总的误差增加．实际上，计算是以浮点形式完成的．因此，数 $hf(y_n, t_n)$ 计算成给定位数的有效数字．但是，将 $hf(y_n, t_n)$ 加到 y_n 上去将引起一个大小为 εy_n 的误差，其中 ε 在 $\left[-\frac{1}{2},\frac{1}{2}\right]10^{-k}$ 范围内，k 为保留的有效数字的位数，假定已经舍入[1]．如果要使方法在实际上收敛，它意味着当 h 减小时，$hf(y_n, t_n)$ 中的准确数字位数增加，

1) 虽然应该这样做，但仍有许多机器不是这样做的．IBM7090 系列机上许多 FORTRAN 配件，例如对乘法用截断解，不是舍入的．由于误差在 $[0,1]10^{-k}$ 范围内，长时间积分的影响可能是显著的．

使得这个误差的值为 $o(h)$[1]，并且计算 $y_n + hf(y_n, t_n)$ 的精度也必须提高，使得 $10^{-k} = o(h)$. 这样，当 h 每减小十倍，必须提高超过一位小数的精度.

幸而舍入误差不是固定的，并且也不是每次都取最坏的情形. 我们常常假定它按均匀分布随机地在其值域中取值，即任何值与任何别的值是同样可能的. 在这个假定下，我们可以讨论有关在若干步上舍入误差影响的一些情况. 如果出现在 (1.22) 中的 $g(t)$ 为零，则对每个舍入误差简单地进行求和可以找到舍入误差的影响. 区域 $\left[-\frac{1}{2}\varepsilon, \frac{1}{2}\varepsilon\right]$ 内 N 个误差的和属于区间 $\left[-\frac{N}{2}\varepsilon, \frac{N}{2}\varepsilon\right]$. 但是，我们还要问"这个和离区间的中点会有多远?"这个问题通常由计算标准偏差来回答，它是 $(R-\mu)^2$ 的均值的平方根，其中 R 为解，而 μ 是 R 的均值. 现在的情形，μ 为0. 如果每步的误差都是相互无关的，则每个都是 $\left[-\frac{\varepsilon}{2}, \frac{\varepsilon}{2}\right]$ 范围内均匀分布的 N 个随机变量之和的标准偏差 $\varepsilon\sqrt{N/12}$. 这个结果告诉我们，虽然误差可以坏到 $N\varepsilon/2$，但是，我们不相信它会比 $\varepsilon\sqrt{N/12}$ 更坏. 事实上，可以证明，对于充分大的 N，误差小于三倍标准偏差的概率为 99.73%[2].

如果舍入误差是偏的，即如果它有不为零的均值，则 N 个这样的误差和的均值为每个均值的 N 倍. 机器若象乘法运算之后截去多余数字 (这叫作截断) 那样截断，可能引起较大量的误差. 因此，IBM7094 机上的乘法产生一个相对于解的均值

---

1) 符号 $o(h)$ 表示 h 的一个函数，它比 h 更快地收敛到零，因此当 $h\to 0$ 时，$\frac{o(h)}{h}\to 0$.

2) 见 P. Henrici (1962), 1.5.2 节.

为 2^{-28} 并在 $[0, 2^{-27}]$ 中取值的误差． N步以后，均值为 $2^{-28}N$，它线性地随着N增加．

表 1.1 中的计算是用 8 位有效数字完成的，所以，舍入误差还不成问题． 表 1.4 说明了只用三位有效数字来完成同样的计算的影响．在一列中，我们看到用截断得到的解(y_T)，而在另一列中，为了得到 y_R，每一步均进行了正确的舍入．这两个结果中的误差在下面二列中列出．我们看到，由于 Euler 方法的截断误差减小，它们开始是减小的，然后随着舍入误差愈来愈大，它们也增加了．实际的截断误差 e 由表 1.1 说明，所以，最后两列表明了舍入误差的作用．

表 1.4. 舍入误差

h	数值解		误差		截断误差 e 由表(1.1)	舍入误差	
	y_T	y_R	$e_T = 2.72 - y_T$	$e_R = 2.72 - y_R$		$e_T - e$	$e_R - e$
$\frac{1}{2}$	2.25	2.25	0.47	0.47	0.47	0	0
$\frac{1}{4}$	2.43	2.44	0.29	0.28	0.28	0.01	0
$\frac{1}{8}$	2.43	2.48	0.29	0.24	0.15	0.14	0.09
$\frac{1}{16}$	2.41	2.55	0.31	0.17	0.08	0.23	0.09
$\frac{1}{32}$	2.33	2.56	0.39	0.16	0.04	0.33	0.12
$\frac{1}{64}$	1.64	1.78	1.08	0.94	0.02	1.06	0.92

1.3.5. 由数值近似产生的扰动

对许多问题，微分方程只能近似地表达一个物理问题．可能一些项忽略了，系数中有误差等．因此，物理系统按

$$w'(t) = f(w(t), t) + s(t),$$
$$w(0) = y(0) + d_0$$

变化，其中 d_0, $s(t)$ 表示由于问题叙述不准确而引起的对方程的扰动．在这一节中，我们将证明数值解等同于具有不同扰动的微分方程的解，即 $y_n = z(t_n)$，这里 z 由

$$z'(t) = f(z(t), t) + r(t),$$
$$z(0) = y(0) + e_0 = y_0$$

给出．而且，选取足够小的 h 和充分高的精度，能使 $\|r(t)\|$ 任意小．因此，由于数值不准确而引起的扰动较之问题不准确而引起的扰动可以任意小．

由于数值解只给定在有限个点的集合上，存在许多不同的函数 $z(t)$，它们在这个集合上与数值解一致．我们对其中使余项 $r(t) = z'(t) - f(z(t), t)$ 为小的 $z(t)$ 感兴趣．

我们用 $y(t; x, \tau)$ 表示 (y, t)-平面上方程 $y' = f(y, t)$ 过点 (x, τ) 的解(我们只处理 f 满足连续性及 Lipschitz 条件且在区域内的点)．我们考虑微分方程过 (y_n, t_n) 的解．如果从 (y_n, t_n) 用 Euler 方法以步长 τ 算一步，我们定义局部误差为

$$d(\tau;\ y_n t_n) = y_n + \tau f(y_n, t_n) - y(t_n + \tau; y_n, t_n). \quad (1.24)$$

它为 Euler 方法的解与过 (y_n, t_n) 的真解之差．换句话说，它为解 $y(t; y_n, t_n)$ 的局部截断误差［见方程(1.8)］．因为我们已经看到，光滑解的局部截断误差象 h^2 那样变化，所以记

$$d(\tau; y_n, t_n) = \tau^2 T(\tau; y_n, t_n) \quad (1.25)$$

并且假定 T 是 τ 的可微函数(如果 y''' 连续，即如此)．

数值方法产生的解为

$$y_{n+1} = y_n + hf(y_n, t_n) + R,$$

其中 R 为舍入误差．现在在区间 $(t_n, t_n + h)$ 中定义[1] $z(t)$ 为

$$z(t_n + \tau) = y_n + \tau f(y_n, t_n) + \frac{\tau R}{h} + \left(\frac{h}{\tau} - 1\right)$$
$$(y_n + \tau f(y_n, t_n) - y(t_n + \tau; y_n, t_n)). \quad (1.26)$$

---

1) 可惜这是一个“人为地构造出来的”函数，然后能够看出它提供了所需要的解．证明从结果回到这个或类似的函数是可能的，但是这样做会使重要的出发点模糊不清．

注意 $z(t_n)=y_n$, $z(t_n+h)=y_{n+1}$, 所以，这样定义的 $z(t)$ 是连续的，通过数值解，并且除了有限个点的集合外是处处可微的. 它是从右边处处可微的，并且它的导数除了一个有限的点的集合外是连续的[1]. 我们考察 $z(t)$ 的余项

$$r(t_n+\tau)=z'(t_n+\tau)-f(z(t_n+\tau),t_n+\tau).$$

将 (1.24), (1.25) 和 (1.26) 代入，得到

$$\begin{aligned}r(t_n+\tau)&=\frac{d}{d\tau}\left[y(t_n+\tau;\,y_n,\,t_n)+\frac{\tau R}{h}+h\tau T(\tau;y_n,\,t_n)\right]\\&\quad-f(z(t_n+\tau),t_n+\tau)\\&=\frac{d}{d\tau}y(t_n+\tau;\,y_n,\,t_n)+\frac{R}{h}+hT(\tau;y_n,\,t_n)\\&\quad+h\tau\frac{d}{d\tau}T(\tau;\,y_n,t_n)-f(z(t_n+\tau),\,t_n+\tau),\end{aligned}$$

它给出

$$\begin{aligned}&r(t_n+\tau)\\&=\frac{R}{h}+hT(\tau;\,y_n,\,t_n)+h\tau\frac{d}{d\tau}T(\tau;\,y_n,\,t_n)\\&\quad+f(y(t_n+\tau;y_n,t_n),t_n+\tau)\\&\quad-f(z(t_n+\tau),t_n+\tau).\end{aligned}\tag{1.27}$$

这样，数值解由微分方程

$$z'(t)=f(z(t),t)+r(t)$$

的解给出，其中 $r(t)$ 由 (1.27) 确定. $r(t)$可用 Lipschitz 条件及导数的界来限定. 注意到对 Euler 方法有

$$\begin{aligned}&\tau^2T(\tau;\,y_n,\,t_n)\\&=y_n+\tau f(y_n,\,t_n)-y(t_n+\tau;y_n,\,t_n)\\&=-\frac{\tau^2}{2}\frac{d^2}{dt^2}y(t_n;\,y_n,\,t_n)-\frac{1}{6}\tau^3\frac{d^3}{dt^3}y(\xi\tau;\,y_n,\,t_n),\end{aligned}\tag{1.28}$$

---

1) 定义一个处处连续可微的函数 $z(t)$ 是可能的，但它只能使定义复杂化，而对分析毫无帮助.

而经过微分得

$$2\tau T(\tau;\ y_n,\ t_n)+\tau^2\frac{dT}{d\tau}(\tau;\ y_n,\ t_n)$$

$$=f(y_n,\ t_n)-\frac{d}{dt}y(t_n+\tau;\ y_n,\ t_n)$$

$$=-\tau\frac{d^2}{dt^2}y(t_n;\ y_n,\ t_n)-\frac{\tau^2}{2}\frac{d^3}{dt^3}y(t_n+\xi_\tau;\ y_n,\ t_n). \qquad (1.29)$$

因此,如果M是y的三阶导数在所考虑的区域中的界,我们将(1.29)乘上h/τ再减去(1.28)乘上h/τ^2, 得到

$$\left|hT(\tau;\ y_n,\ t_n)+h\tau\frac{d}{d\tau}T(\tau;\ y_n,\ t_n)+\frac{hy_n''}{2}\right|\leqslant\frac{2}{3}h\tau M,$$

其中 $y_n''=(d^2/dt^2)y(t_n;\ y_n,\ t_n)$. 如果$T$为$T(\tau;\ y_n,\ t_n)$对$\tau\leqslant h$的界, 则

$$|f(y(t_n+\tau;y_n,t_n),\ t_n+\tau)-f(z(t_n+\tau),\ t_n+\tau)|\leqslant LT\tau^2,$$

并且由(1.27)得到

$$\left|r(t_n+\tau)-\frac{R}{h}+\frac{h}{2}y_n''\right|\leqslant LT\tau^2+\frac{2}{3}h\tau M$$

$$\leqslant\left(LT+\frac{2}{3}M\right)h^2. \qquad (1.30)$$

这表明$r(t)$实质上是h^{-1}阶的(舍入误差加截断误差), 虽然在这种情形下,二阶导数y''是在数值解y_n上求值,而不是象(1.20)中所做的那样在$y(t_n)$上求值.

问　　题

1. 下面给出的初值问题是适定的吗?

   (a) $y'=\sqrt{1-y^2}$, $y(0)=0$.

   (b) $y'=+\sqrt{y^2-1}$, $y(0)=2$.

2. 用 Euler 方法从$t=0$到$t=1$积分方程$y'=2t$, 要使误差不超

过下面的数值：

(a) 0.1;

(b) 0.01;

(c) 0.001,

应该用多大的步长？

3. 假定积分所用的机器对所有数字截断到 10^{-6}（不是舍人）. 用步长为 h 的 Euler 方法积分方程

$$y' = -20(y - t^2) + 2t, \quad y(0) = 0,$$

粗略画出和描述 $\log(t = 100$ 时的误差$)$ 对 $\log(h)$ 的曲线的主要特征，h 在 0.0001 到 1 的范围中变化.

4. 从 $t = 0$ 到 $t = 1$ 用 Euler 方法积分下列方程：

(a) $dy/dt = 1 + 3t^2 - y + t^3, \quad y(0) = 1;$

(b) $dy/dt = 2t + 1000(2 + t^2) - 1000y, \quad y(0) = 2,$

为了达到误差小于 0.05，不考虑舍人误差，要用多大的步长？

5. 当机器的曲线显示器连续画出一个圆时，x, y 的逐次值可以用

$$x = r\mathrm{cso}\theta,$$
$$y = r\sin\theta,$$
$$\theta = nh(n = 0, 1, \cdots)$$

计算. 这样做非常消耗时间，我们可用解微分方程

$$\frac{dx}{d\theta} = -r\sin\theta = -y, \quad x(0) = r,$$

$$\frac{dy}{d\theta} = r\cos\theta = x, \quad y(0) = 0$$

来代替. 如果用 Euler 方法，得到

$$x_{n+1} = x_n - hy_n,$$
$$y_{n+1} = y_n + hx_n,$$

这是一个坏的方法. 事实上，它画出一条螺线. 为什么？如果改用下面的公式

$$x_{n+1} = x_n - hy_n,$$
$$y_{n+1} = y_n + hx_{n+1},$$

点就组成一个闭合形. 为什么？在讨论中，舍人误差有些什么影响？

6. 对下列问题:

(a) $y' = 2ty$;

(b) $y' = -2ty$;

(c) $y' = (2 + t^2 - y^2)^{1/2}$;

(d) $y' = t^{1/2}$;

(e) $y' = t\sqrt{y - 1 + 3t^3}$

推导一些误差的界;它们都在[0, 1]上以步长 h 积分,初值 $y(0)=1$。将这些界与同一问题的渐近误差估计进行比较。

2. 高阶单步方法

第 1 章讨论的 Euler 方法称为单步方法，因为这个算法描述了这样一种数值技术：只利用前一步的值，即 t_n 的值来计算解在 t_{n+1} 点上的近似值。我们知道，Euler 方法的误差，当 $h \to 0$ 时按 $O(h)$ 变化。由于这个原因，我们把它称为一阶方法。对于某个 $r > 0$，如果有一个方法，其误差按 $O(h^r)$ 变化，那么，我们就称它为 r 阶方法。注意，在 Euler 方法中，局部截断误差是 $O(h^2)$，比全体误差（即在整个区域上的误差）高 h 的一次方。在第 4 章我们将证明，r 阶方法的局部截断误差等于 $O(h^{r+1})$，或者说全体误差是 $O(h^{-1} \times$ 局部截断误差)。定理的证明放到第 4 章，并且这一章的讨论只限于比 Euler 方法更高阶的单步方法的实用方面。

我们首先要问，为什么对高阶方法感兴趣。详细的讨论放到第 5 章，那时我们才具备必要的预备知识。这里的简单回答是指出：如果 h 很小，则 h^2 就更小，这样用小步长 h 的高阶方法就能达到更高的精确度，并且当 h 趋于零时，高阶方法收敛得更快。

2.1. Taylor 级数方法

Euler 方法可以看作是 Taylor 级数前两项的近似。如果能计算 y 的高阶导数，则可写出

$$y_{n+1} = y_n + hy_n' + \frac{h^2}{2} y_n'' + \cdots + \frac{h^r}{r!} y_n^{(r)}. \qquad (2.1)$$

我们有 $y_n' = f(y_n, t_n)$，所以，能按下面那样计算高阶导数。把

$f(y, t)$, $\partial f/\partial t$, $\partial f/\partial y$ 分别写成 f, f_t, f_y, 于是，对于下面两个导数，有

$$y'' = f_t + f_y f,$$
$$y''' = f_{tt} + f_{ty} f + f_{yt} f + f_y f_t + f_{yy} f^2 + f_y^2 f$$
$$= f_{tt} + 2f_{ty} f + f_y f_t + f_{yy} f^2 + f_y^2 f. \qquad (2.2)$$

显然这不是一个实用的方法，除非函数 f 足够简单，使得这些偏导数中许多为零. 但是，必要时推导如 (2.2) 的许多公式以及为了代入 (2.1) 数值计算这些导数，在理论上是可能的. 局部截断误差是

$$\frac{h^{r+1} y^{(r+1)}}{(r+1)!}.$$

手算相当复杂的函数 $f(y, t)$ 的微分是麻烦的，并且容易出错. 可以应用符号微分[1]的计算技术，虽然可能得到对计算来说是太长的非常复杂的公式. 一般来说，在计算机上求解 Taylor 级数的方法是不实用的，但是，对于简单问题得到较低精度的近似值(手算或使用台式机计算)，还是比较有用的. 如果函数 $f(y, t)$ 只能用复杂的子程序来计算，而且用这种子程序求微分实际上是不可能的，那么，这种方法是没有用的.

2.2. Richardson 外插法 ($h = 0$)

我们知道，Euler 方法给出了一个形如 $h\delta(b) + O(h^2)$ 的误差，如果不计[2] 舍入误差，其中 $\delta(b)$ 仅依赖于微分方程. 为了把 Euler 方法引起的误差减半，如果忽略 $O(h^2)$ 项，就必须

1) 符号微分，在计算机中函数 f 按其代数形式表示为一序列符号(字符). 微分运算生成一个表示对于所给定变量的导数的代数形式的一种新的符号序列. 详述，参见 Engeli (1969).

2) 根据这点，在讨论中不计舍入误差，假定在每一步它们都小于局部截断误差. 如果不是这样，则适当换算包含舍入误差在内的局部截断误差就可估计舍入误差的影响.

把步长减半．我们考虑一个微分方程的两个积分，一个用步长 h，另一个用步长 $h/2$．若结果分别是 $y_h(b)$ 和 $y_{h/2}(b)$，则可写出

$$\begin{cases} y_h(b) = y(b) + h\delta(b) + O(h^2), \\ y_{h/2}(b) = y(b) + \dfrac{h}{2}\delta(b) + O(h^2). \end{cases} \tag{2.3}$$

由 (2.3) 消去 $\delta(b)$，得到

$$y(b) = 2y_{h/2}(b) - y_h(b) + O(h^2). \tag{2.4}$$

我们可以利用这个结果作为较好的数值近似．表 2.1 表明了这一点．第 2 列是以步长 $2^{-p}(p = 0, 1, 2, 3, 4, 5, 6)$ 用 Euler 方法积分 $y' = -y$ 的结果，如同表 1,2 得到的结果一样．第 3 列是由方程 (2.4) 形成一个较好的近似获得的．我们看到新的近似公式比 Euler 公式更精确，而且当 h 减小时，更加迅速地满足精确度，因为误差是 $O(h^2)$ 而不是 $O(h)$．这个过程称为对极限的延伸逼近，这个新的公式给出一个二阶方法，误差为 $O(h^2)$．

表 2.1. Richardson 外播法

h	Euler $y_h(1)$	$2y_{h/2}(1) - y_h(1)$	误差	误差 $/h^2$
1	0.0000000	0.5000000	0.1321206	0.1321206
$\frac{1}{2}$	0.2500000	0.3828125	0.0149331	0.0597322
$\frac{1}{4}$	0.3164063	0.3708116	0.0029321	0.0469141
$\frac{1}{8}$	0.3435089	0.3685393	0.0006599	0.0422317
$\frac{1}{16}$	0.3560741	0.3680364	0.0001569	0.0401740
$\frac{1}{32}$	0.3620552	0.3679177	0.0000382	0.0391373
$\frac{1}{64}$	0.3649865	—	—	—

2.3. 二阶 Runge-Kutta 方法

当应用到单个步长 h 时，我们更仔细地考察上节的近似公式．我们用一个步长 h，得到

$$y_h(t_n + h) = y_n + hf(y_n, t_n),$$

以及用二个步长 $\frac{h}{2}$，有

$$q_1 = y_{h/2}\left(t_n + \frac{h}{2}\right) = y_n + \frac{h}{2} f(y_n, t_n),$$

$$y_{h/2}(t_n + h) = q_1 + \frac{h}{2} f\left(q_1, t_n + \frac{h}{2}\right),$$

于是

$$\begin{aligned} & y(t_n + h) \\ &= 2y_{h/2}(t_n + h) - y_h(t_n + h) + O(h^2) \\ &= 2q_1 + hf\left(q_1, t_n + \frac{h}{2}\right) - y_n - hf(y_n, t_n) + O(h^2). \end{aligned}$$

实际所包括的计算是

$$\begin{cases} q_1 = y_n + \frac{h}{2} f(y_n, t_n), \\ y_{n+1} = y_n + hf\left(q_1, t_n + \frac{h}{2}\right). \end{cases} \tag{2.5}$$

这个方法的形式与 Euler 方法类似，是将某些量加到该步的起始值上而得到该步的终了值．这个方法称为中点方法．若记加上去的量为 $h\phi(y, t, h)$，则有

$$y_{n+1} = y_n + h\phi(y_n, t_n, h), \tag{2.6}$$

其中

$$\phi(y_n, t_n, h) = f\left(y_n + \frac{h}{2} f(y_n, t_n), t_n + \frac{h}{2}\right).$$

显然，我们希望 $h\phi$ 近似于

$$hy'_n + \frac{h^2}{2} y''_n + \frac{h^3}{6} y'''_n + \cdots,$$

以便使 (2.6) 近似于 t_n 处的 Taylor 展式．若假定 $f(y, t)$ 具有充分可微的条件，则可按下面的办法估计 (2.6) 与 Taylor 级数之间的一致项：

$$q_1 = y_n + \frac{h}{2} f(y_n, t_n) = y_n + \frac{h}{2} y'_n,$$

$$y_{n+1} = y_n + hf\left(y_n + \frac{h}{2} y'_n, t_n + \frac{h}{2}\right)$$

$$= y_n + hf(y_n, t_n) + \frac{h^2}{2} y'_n f_y(y_n, t_n)$$

$$+ \frac{h^2}{2} f_t(y_n, t_n) + O(h^3)$$

$$= y_n + hy'_n + \frac{h^2}{2}(f_y y' + f_t)_n + O(h^3).$$

但是

$$y'' = f_t + f_y y',$$

因而

$$y_{n+1} = y_n + hy'_n + h^2/2 y''_n + O(h^3).$$

于是,(2.6)的局部截断误差为 $O(h^3)$.我们可在每一步上重复使用公式(2.6)而不使用 Richardson 外插法. 在表 2.2 中, 用步长 $h = 0.25$ 积分方程 $y' = -y$, $y(0) = 1$, 说明了这个方法.

表 2.2. 中点方法

t_n	y_n	$\frac{h}{2} f(t_n, y_n)$	$hf\left(y_n + \frac{h}{2} f(t_n, y_n), t_n + \frac{h}{2}\right)$
0.00	1	−0.125	−0.21875
0.25	0.78125	−0.0976563	−0.1708984
0.50	0.6103516	−0.0762940	−0.1335144
0.75	0.4768372	−0.0596046	−0.1043081
1.00	0.3725291	—	—

推导方程(2.5)的另外一个途径是回到 Taylor 级数方法, 并且提出如下问题: 如何形成

$$y_{n+1} = y_n + hy'_n + \frac{h^2}{2} y''_n$$

而得到二阶方法．若考察 $hf\left(y\left(t_n+\frac{h}{2}\right),t_n+\frac{h}{2}\right)$，即考虑 $hy'(t_n)+[h^2y''(t_n)/2]+O(h^3)$．虽然不知道 $y\left(t_n+\frac{h}{2}\right)$的值，仍可用 Euler 方法近似，而得到

$$y\left(t_n+\frac{h}{2}\right)=y(t_n)+\frac{h}{2}y'(t_n)+O(h^2).$$

由此得到

$$\begin{aligned}&y(t_n)+hy'(t_n)+\frac{h^2}{2}y''(t_n)\\&=y(t_n)+hf\left(y(t_n)+\frac{h}{2}y'(t_n),t_n+\frac{h}{2}\right)+O(h^3).\end{aligned}$$

当以 y_n 代替 $y(t_n)$ 时，得到与 (2.6) 同样的方程．

也可用其他近似公式逼近 Taylor 级数的项，例如，有

$$\frac{hy'(t_n)+hy'(t_{n+1})}{2}=hy'(t_n)+\frac{h^2}{2}y''(t_n)+O(h^3).$$

这样，可写

$$\begin{aligned}y(t_{n+1})-y(t_n)=\frac{h}{2}(&f(y(t_n),t_n)\\&+f(y(t_{n+1}),t_{n+1}))+O(h^3).\end{aligned}\tag{2.7}$$

若忽略 $O(h^3)$ 项且用 (2.7) 计算 $y(t_{n+1})$ 的近似值，则称为梯形方法．因为 $y(t_{n+1})$ 未知，所以 (2.7) 右端无法计算．但有两种可用的方法：一种是想法解 $y(t_{n+1})$ 的非线性方程 (2.7)，这称为隐式方法；另一种是用其他方法(例如 Euler 方法)估算 $y(t_{n+1})$，而得到显式方法．因此，我们得到如下公式：

$$\begin{cases}\bar{y}(t_{n+1})=y(t_n)+hf(y(t_n),t_n)+O(h^2),\\y(t_{n+1})=y(t_n)+\frac{h}{2}(f(y(t_n),t_n)\\\qquad\qquad+f(\bar{y}(t_{n+1}),t_{n+1}))+O(h^3).\end{cases}\tag{2.8}$$

它称为改进的梯形方法或 Heun 方法．我们不详细分析这个方法，而考察一类二阶显式方法．考虑

$$q_1 = y_n + \alpha h f(y_n, t_n),$$
$$y_{n+1} = y_n + \beta h f(y_n, t_n) + \gamma h f(q_1, t_n + \eta h). \quad (2.9)$$

我们希望使得 y_{n+1} 的展式尽可能与 Taylor 级数一致. 由 (2.9)

$$y_{n+1} = y_n + \beta h y_n' + \gamma h y_n' + \alpha\gamma h^2 f_y y_n' + \gamma\eta h^2 f_t + \frac{\alpha^2}{2}\gamma h^3 f_{yy}(y_n')^2 + \alpha\gamma\eta h^3 f_{yt} y_n' + \frac{\gamma}{2}\eta^2 h^3 f_{tt} + O(h^4).$$

整理诸项使之与 $t = t_n$ 的 Taylor 级数对应项一致, 得到

$$(\beta + \gamma) h y_n' = h y_n',$$
$$\gamma(\alpha f_y y_n' + \eta f_t) h^2 = \frac{h^2}{2} y_n'' = \frac{h^2}{2}(f_y y_n' + f_t).$$

因此

$$\begin{cases} \beta = 1 - \gamma, \\ \alpha = \eta = \dfrac{1}{2\gamma}. \end{cases} \quad (2.10)$$

误差项趋于零的渐近性质, 由考察 Taylor 级数的后一项来确定. 方法的 h^3 项与 $h^3 y_n'''/6$ 之间的差是

$$\frac{h^3}{6}(3\alpha^2\gamma f_{yy}(y_n')^2 + 6\alpha\gamma\eta f_{yt} y_n' + 3\gamma\eta^2 f_{tt} - y_n''')$$
$$= \frac{h^3}{6}\left[\frac{3}{4\gamma}(f_{yy}(y_n')^2 + 2 f_{yt} y_n' + f_{tt}) - y_n'''\right]. \quad (2.11)$$

取 $\gamma = 1$, 得到中点方法 (2.5); 取 $\gamma = \frac{1}{2}$, 给出 Heun 方法[1]. 如果代入到 (2.9) 的展式已经包含了 Taylor 级数的 $O(h^3)$ 余项, 那么, 将得到一个与 (2.11) 类似的表达式, 而且可用它来肯定误差小于形如 Mh^3 的表达式. 在类似于对 Euler 方法的定理 1.2 给出的误差界中可以使用这一点. 等价定理将在第 4 章讨论.

1) 这些二阶方法的名称在文献中是不一致的. 对于不同的 γ 值, 不同的作者使用不同的名称, 表 2.3 指出了这一点.

用 (2.9) 和 (2.10) 所描述的方法，称为二阶 Runge-Kutta 方法．我们可以选取自由参数 γ 的值，使得方法的某个需要的性质达到最佳．关于 γ 的特殊标准是由 Ralston (1965) 给出的，见 § 5.6.3.1．(2.11) 中出现的导数都规定了界，并且选择 γ 使 (2.11) 的界减至最小．由 Lotkin (1951) 提出的这个导数的假定界限是

$$\left|\frac{\partial^{i+j}}{\partial t^i \partial y^j} f\right| \leqslant \frac{L^{i+j}}{M^{j-1}}, \qquad |f(y, t)| \leqslant M, \qquad (2.12)$$

所以，(2.11) 能小于

$$|\text{误差}| = \frac{h^3}{6}\left|\left[\left(\frac{3}{4\gamma}-1\right)[f_{yy}f^2 + 2f_{yt}f + f_{tt}] - f_y^2 f - f_y f_t\right]\right|$$

$$\leqslant \frac{h^3}{6}\left[\left|\frac{3}{4\gamma}-1\right| 4ML^2 + 2ML^2\right].$$

当 $\gamma = \dfrac{3}{4}$ 时，它有极小值 $h^3ML^2/3$.

2.4. 显式 Runge-Kutta 方法

前节我们推导了一类二阶方法，推导方法如下：首先，在点 $t_n + \alpha h$ 进行近似求解，然后利用这点的 $f(y, t)$ 值与其在 t_n 的值一道匹配 Taylor 级数的项．本节将推广这类方法，首先，在若干个附加点 $t_n + \alpha_i h$ 上进行近似求解，然后与 Taylor 级数的更多的项一致来得到高阶方法．在开始进行这个工作之前，为了方便起见，引进一些处理导数的简化记号．

定义 f 为具有分量 $f^0 = 1$ 和 $f^1 = f(y, t)$ 的向量，于是 f^0 和 f^1 分别是 t 和 y 对于 t 的导数．记 y 关于 t 的二阶导数为

$$y'' = (f^1)' = \frac{\partial f}{\partial t}\frac{dt}{dt} + \frac{\partial f^1}{\partial y}\frac{dy}{dt} = \frac{\partial f^1}{\partial t} f^0 + \frac{\partial f^1}{\partial y} f^1.$$

若将 $\partial z/\partial t$, $\partial z/\partial y$ 分别记为 z_0, z_1, 则得到

表 2.3. 普通的二阶 Runge-Kutta 方法

作　　者	$\gamma=\frac{1}{2}$	$\gamma=\frac{3}{4}$	$\gamma=1$
Ceschino 和 Kuntzman(1966)	Euler-Cauchy	Heun	改进切线
Collatz (1960)	改进多边形		改进多边形
Henrici (1962)	Heun		改进多边形或改进 Euler
Isaacson 和 Keller (1962)	改进 Euler	Heun	Euler-Cauchy

Heun (1900) 讨论了一类方法，对这些方法中的任何一个确定其名称好像是困难的.

$$y'' = f^1_0 f^0 + f^1_1 f^1 = \sum_{i=0}^{1} f^1_i f^i.$$

最后，利用求和的习惯记法，即在乘积项中，任何重复的下标或上标是指在它们的变化范围内求和，在现在的情形，范围是 0 和1．于是 $a_i b^i$ 是 $\Sigma^1_{i=0} a_i b^i$ 的缩写，而 $a_i b^i_j c^j$ 是

$$\sum_{i=0}^{1}\sum_{j=0}^{1} a_i b^i_j c^j$$

的缩写(这种约定仅在本章使用)．应用这种记号，有

$$y'' = f^1_i f^i. \tag{2.13}$$

若定义 $y^0 = t$, $y^1 = y$, 则可记

$$(y^i)'' = f^i_j f^j.$$

注意 z 关于 t 的微分法则是 $(z)' = z_i f^i$．所以，可得到

$$(y^i)''' = ((y^i)'')' = (f^i_j f^j)' = f^i_{jk} f^j f^k + f^i_j f^j_k f^k \tag{2.14}$$

和

$$(y^i)^{(4)} = (f^i_{jk} f^j f^k + f^i_j f^j_k f^k)' = f^i_{jkl} f^j f^k f^l + f^i_{jk} f^j_l f^k f^l$$
$$+ f^i_{jk} f^j f^k_l f^l + f^i_{jl} f^j_k f^k f^l + f^i_j f^j_{kl} f^k f^l + f^i_j f^j_k f^k_l f^l.$$

现在注意

$$f^i_{jk} = \frac{\partial^2 f^i}{\partial y^j \partial y^k} = \frac{\partial^2 f^i}{\partial y^k \partial y^j} = f^i_{kj},$$

即下标的次序是不重要的，而且由于 j，k 和 l 在求和中是下标的记号，故有

$$f^i_{jk}f^jf^k_lf^l = f^i_{kj}f^kf^j_lf^l \ (j \text{ 和 } k \text{ 互换})$$
$$= f^i_{jk}f^j_lf^lf^k (\text{改变微分次序}).$$

这样，$(y^i)^{(4)}$ 的第二项和第三项是相同的．类似地，第四项是

$$f^i_{jl}f^j_kf^kf^l = f^i_{jk}f^j_lf^lf^k \ (l \text{ 和 } k \text{ 互换}),$$

于是

$$(y^i)^{(4)} = f^i_{jkl}f^jf^kf^l + 3f^i_{jk}f^j_lf^kf^l + f^i_jf^j_{kl}f^kf^l + f^i_jf^j_kf^k_lf^l. \tag{2.15}$$

利用这个记号，考虑再增加一个中间点或者中间级的(2.9)的自然推广，并且研究三级 (three-stage) Runge-Kutta 方法．记

$$\begin{aligned}
k^i_0 &= hf^i(y_n), \\
k^i_1 &= hf^i(y_n + \alpha_1 k_0), \\
k^i_2 &= hf^i(y_n + \beta_{21}k_0 + (\alpha_2 - \beta_{21})k_1) \\
&= hf^i(y_n + \alpha_2 k_0 + (\alpha_2 - \beta_{21})(k_1 - k_0)), \\
y^i_{n+1} &= y^i_n + \gamma_0 k^i_0 + \gamma_1 k^i_1 + \gamma_2 k^i_2.
\end{aligned} \tag{2.16}$$

注意，我们没有使 f 明显依赖于 t，因为 y_n 可取为包含因变量和自变量作为分量的一个向量．当然，$f^0(y_n + \alpha k) = 1$，这样，k^0_p 就是 h．当 $y_n = y(t_n)$ 时，我们要求由 (2.16) 确定的 y_{n+1} 与 $y(t_{n+1})$ 的 Taylor 级数尽可能多的项一致．由 (2.13) 到 (2.15) 得到

$$\begin{aligned}
y^i(t_{n+1}) = y^i + hf^i + \frac{h^2}{2}(f^i_jf^j) + \frac{h^3}{6}(f^i_{jk}f^jf^k + f^i_jf^j_kf^k) \\
+ \frac{h^4}{24}(f^i_{jkl}f^jf^kf^l + 3f^i_{jk}f^j_lf^lf^k + f^i_jf^j_{kl}f^kf^l \\
+ f^i_jf^j_kf^k_lf^l) + O(h^5),
\end{aligned} \tag{2.17}$$

式中右端的每个值都在 $t = t_n$ 计算．比较 (2.16) 和 Taylor 级数展式，得到

$k^i_0 = hf^i$,

$$k_1^i = hf^i + \alpha_1 h^2 f_j^i f^j + \frac{\alpha_1^2 h^3}{2} f_{jk}^i f^j f^k + \frac{\alpha_1^3 h^4}{6} f_{jkl}^i f^j f^k f^l + O(h^5),$$

$$\begin{aligned}
k_2^i = {} & hf^i + \alpha_2 h^2 f_j^i f^j + \frac{\alpha_2^2 h^3}{2} f_{jk}^i f^j f^k + \frac{\alpha_2^3 h^4}{6} f_{jkl}^i f^j f^k f^l \\
& + (\alpha_2 - \beta_{21}) h f_j^i (k_1^j - k_0^j) \\
& + \alpha_2(\alpha_2 - \beta_{21}) h^2 f_{jk}^i f^j (k_1^k - k_2^k) + O(h^5) \\
= {} & hf^i + \alpha_2 h^2 f_j^i f^j + h^3 \left[\frac{\alpha_2^2}{2} f_{jk}^i f^j f^k + (\alpha_2 - \beta_{21}) \alpha_1 f_j^i f_k^j f^k \right] \\
& + h^4 \left[\frac{\alpha_2^3}{6} f_{jkl}^i f^j f^k f^l + (\alpha_2 - \beta_{21}) \frac{\alpha_1^2}{2} f_j^i f_{kl}^j f^k f^l \right. \\
& \left. + \alpha_2(\alpha_2 - \beta_{21}) \alpha_1 f_{jk}^i f^j f_l^k f^l \right] + O(h^5).
\end{aligned}$$

由(2.16)得到 y_{n+1}，并令其与(2.17)相等，我们得到含 h, h^2, h^3, 的项如下:

$$h:\ f^i = (\gamma_0 + \gamma_1 + \gamma_2) f^i,$$

$$h^2:\ \frac{1}{2} f_j^i f^j = \gamma_1 \alpha_1 f_j^i f^j + \gamma_2 \alpha_2 f_j^i f^j,$$

$$\begin{aligned}
h^3:\ \frac{1}{6}(f_{jk}^i f^j f^k + f_j^i f_k^j f^k) = {} & \frac{1}{2} \gamma_1 \alpha_1^2 f_{jk}^i f^j f^k \\
& + \frac{1}{2} \gamma_2 \alpha_2^2 f_{jk}^i f^j f^k + \gamma_2 (\alpha_2 - \beta_{21}) \alpha_1 f_j^i f_k^j f^k.
\end{aligned}$$

因为偏导数是任意的,故每个不同的组合必定分别为零,于是得到方程

$$\begin{aligned}
1 &= \gamma_0 + \gamma_1 + \gamma_2, \\
\frac{1}{2} &= \gamma_1 \alpha_1 + \gamma_2 \alpha_2, \\
\frac{1}{6} &= \frac{1}{2} \gamma_1 \alpha_1^2 + \frac{1}{2} \gamma_2 \alpha_2^2, \\
\frac{1}{6} &= \gamma_2 (\alpha_2 - \beta_{21}) \alpha_1.
\end{aligned}$$

这是六个未知数四个方程的方程组，它有用 α_1 和 α_2 表示的双参数解族:

$$\gamma_1 = \frac{3\alpha_2 - 2}{6\alpha_1(\alpha_2 - \alpha_1)},$$

$$\gamma_2 = \frac{3\alpha_1 - 2}{6\alpha_2(\alpha_1 - \alpha_2)},$$

$$\gamma_0 = 1 - \gamma_1 - \gamma_2,$$

$$\beta_{21} = \alpha_2 - \frac{1}{6\alpha_1\gamma_2}.$$

误差项 $y_{n+1}^i - y^i(t_{n+1})$如下:

$$\begin{aligned} &-\frac{h^4}{24}[f_{jkl}^i f^j f^k f^l(1 - 4\gamma_1\alpha_1^3 - 4\gamma_2\alpha_2^3) \\ &+ f_{jk}^i f_l^j f^l f^k(3 - 24\gamma_2\alpha_1\alpha_2[\alpha_2 - \beta_{21}]) \\ &+ f_j^i f_{kl}^j f^k f^l(1 - 12\gamma_2\alpha_1^2(\alpha_2 - \beta_{21})) \\ &+ f_j^i f_k^j f_l^k f^l] + O(h^5). \end{aligned} \tag{2.18}$$

我们看到，选择参数 α_1 和 α_2 不会使得公式的末项为零，因此三级方法的最大阶是三. 实际上，考虑（$r+1$）阶导数中出现的项 $f_j^i f_k^j \cdots f_y^x f^y$，就容易推广这个结论，即证明 r 级方法的最大的阶是 r. 对于最大阶的附加限制，后面还要讨论.

一般的 r 级显式 Runge-Kutta 方法，由

$$k_0^i = hf^i(y_n),$$

$$k_q^i = hf^i\left(y_n + \sum_{j=1}^{q} \beta_{qj} k_{j-1}\right), \qquad q = 1, 2, \cdots, r-1$$

$$y_{n+1}^i = y_n^i + \sum_{q=0}^{r-1} \gamma_q k_q^i,$$

给出. 这个方法之所以称为显式方法，是因为上述方程每一个的右端只利用前面计算的值就可计算. 未知数 γ_q 和 β_{qj} 的方程（$1 \leqslant j \leqslant q < \gamma$），可利用与对应的 Taylor 级数的项一致的方法来推导.

2.4.1. 经典的 Runge-Kutta 方法

通常的 Runge-Kutta 方法是一个四阶的四级方法，我们可以在 $\alpha_2 - \beta_{21} = \beta_{22}$ 的情形下展开 k_0^i，k_1^i 和 k_2^i，并且能对 k_3^i 作出类似的展开式（这里 $k_3^i = hf^i(y_n + \beta_{31}k_0 + \beta_{32}k_1 + (\alpha_3 - \beta_{31} - \beta_{32})k_2)$），使直到 h 四次方的系数相等．我们得到方程

$$
\begin{aligned}
\gamma_0 + \gamma_1 + \gamma_2 + \gamma_3 &= 1, \\
\gamma_1\alpha_1 + \gamma_2\alpha_2 + \gamma_3\alpha_3 &= \frac{1}{2}, \\
\gamma_1\alpha_1^2 + \gamma_2\alpha_2^2 + \gamma_3\alpha_3^2 &= \frac{1}{3}, \\
\gamma_1\alpha_1^3 + \gamma_2\alpha_2^3 + \gamma_3\alpha_3^3 &= \frac{1}{4}, \\
\gamma_2\alpha_1\beta_{22} + \gamma_3(\alpha_1\beta_{32} + \alpha_2\beta_{33}) &= \frac{1}{6}, \\
\gamma_2\alpha_1\alpha_2\beta_{22} + \gamma_3\alpha_3(\alpha_1\beta_{32} + \alpha_2\beta_{33}) &= \frac{1}{8}, \\
\gamma_2\alpha_1^2\beta_{22} + \gamma_3(\alpha_1^2\beta_{32} + \alpha_2^2\beta_{33}) &= \frac{1}{12}, \\
\gamma_3\alpha_1\beta_{22}\beta_{33} &= \frac{1}{24},
\end{aligned}
\tag{2.19}
$$

其中 $\alpha_q = \sum_{j=1}^{q}\beta_{qj}$，$q = 1, 2, 3$. Ralston (1965, p. 199) 给出了用 α_1 和 α_2 表示的方程 (2.19) 的双参数解族：

$$
\begin{aligned}
\gamma_0 &= \frac{1}{2} + \frac{1 - 2(\alpha_1 + \alpha_2)}{12\alpha_1\alpha_2}, \\
\gamma_1 &= \frac{2\alpha_2 - 1}{12\alpha_1(\alpha_2 - \alpha_1)(1 - \alpha_1)}, \\
\gamma_2 &= \frac{1 - 2\alpha_1}{12\alpha_2(\alpha_2 - \alpha_1)(1 - \alpha_1)}, \\
\gamma_3 &= \frac{1}{2} + \frac{2(\alpha_1 + \alpha_2) - 3}{12(1 - \alpha_1)(1 - \alpha_2)},
\end{aligned}
$$

$$\beta_{22}=\frac{\alpha_2(\alpha_2-\alpha_1)}{2\alpha_1(1-2\alpha_1)},\qquad \alpha_3=1,$$

$$\beta_{32}=\frac{(1-\alpha_1)[\alpha_1+\alpha_2-1+(2\alpha_2-1)^2]}{2\alpha_1(\alpha_2-\alpha_1)[6\alpha_1\alpha_2-4(\alpha_1+\alpha_2)+3]},$$

$$\beta_{33}=\frac{(1-2\alpha_1)(1-\alpha_1)(1-\alpha_2)}{\alpha_2(\alpha_2-\alpha_1)[6\alpha_1\alpha_2-4(\alpha_1+\alpha_2)+3]}. \tag{2.20}$$

在历史上,这个方法首先用于

$$k_0=hf(y_n)$$

$$k_1=hf\left(y_n+\frac{1}{2}k_0\right)$$

$$k_2=hf\left(y_n+\frac{1}{2}k_1\right)$$

$$k_3=hf(y_n+k_2)$$

$$y_{n+1}=y_n+\frac{1}{6}(k_0+2k_1+2k_2+k_3) \tag{2.21}$$

的解,这是通过令 $\alpha_1=\alpha_2=\frac{1}{2}$ 得到的. 我们称此为经典的 Runge-Kutta 公式.

经典的 Runge-Kutta 公式可以看成试图把 Simpson 求积法则:

$$\int_{t_n}^{t_{n+1}}f(x)dt\cong\frac{h}{6}\left(f(t_n)+4f\left(t_n+\frac{h}{2}\right)+f(t_{n+1})\right)$$

推广到微分方程. 如果 $f(y, t)$仅仅是 t 的函数,那么Simpson求积法则与经典的 Runge-Kutta 公式是相等的.

2.4.2. Ralston Runge-Kutta方法

Ralston (1965, p. 200) 对于 $f(y,t)$ 的偏导数, 采用了一个由 (2.12) 给出的形式的界,并且在 (2.19) 中选择两个自由参数,使得求解误差的界限减至最小. 结果是对应于

$$\alpha_1 = 0.4 \text{ 和 } \alpha_2 = \frac{7}{8} - \frac{3}{16}\sqrt{5}$$

的方法，并且由

$$\begin{aligned}
k_0 &= hf(y_n), \\
k_1 &= hf(y_n + 0.4k_0), \\
k_2 &= hf(y_n + 0.29697760k_0 + 0.15875966k_1), \\
k_3 &= hf(y_n + 0.21810038k_0 - 3.05096470k_1 \\
&\quad + 3.83286432k_2), \\
y_{n+1} &= y_n + 0.17476028k_0 - 0.55148053k_1 \\
&\quad + 1.20553547k_2 + 0.17118478k_3
\end{aligned} \tag{2.22}$$

给出.

2.4.3. Butcher 关于 Runge-Kutta 方法可达到阶的结果

至此，已经知道显式 Runge-Kutta 方法的阶和右函数计算次数两者之间的如下关系：

表 2.4. 一到四级 Runge-Kutta 方法的最大阶数

右函数计算的次数 $=\nu$	最大阶数 $=r(\nu)$
1	1
2	2
3	3
4	4
?	?

我们已经看到

$$r(\nu) \leqslant \nu$$

[见表达式(2.18)下面的说明]，自然要问："高阶方法是否存在？" Butcher (1965) 充分研究了 $r(\nu)$，并且证明了显式 Runge-Kutta 方法的如下关系：

表 2.5. 不同级数 Runge-Kutta 方法的最大阶数

ν	$r(\nu)$
5	4
6	5
7	6
$\nu\geqslant 8$	$r(\nu)\leqslant\nu-2$

所有这种类型的方法均称为 Runge-Kutta 方法,虽然原始的方法只是经典的四阶方法.

2.5.隐式 Runge-Kutta 方法

前节讨论的 Runge-Kutta 方法,具有这样的性质,其右端可直接计算.若去掉这个条件,则得到隐式 Runge-Kutta 方法.一个 r 级隐式 Runge-Kutta 方法形如

$$k_q = hf(y_n + \beta_{qj}k_j) \qquad q = 1, 2, \cdots, r,$$
$$y_{n+1} = y_n + \gamma_j k_j \ (j \text{ 从 } 1 \text{ 到 } r \text{ 求和}). \tag{2.23}$$

二级隐式 Runge-Kutta 方法的例子由

$$k_1 = hf(y_n),$$
$$k_2 = hf\left(y_n + \frac{1}{2}k_1 + \frac{1}{2}k_2\right),$$
$$y_{n+1} = y_n + \frac{1}{2}k_1 + \frac{1}{2}k_2$$

给出. 当然，这是梯形法则. 现在我们研究一般的二级隐式 Runge-Kutta 方法:

$$k_1 = hf(y_n + \beta_{11}k_1 + \beta_{12}k_2),$$
$$k_2 = hf(y_n + \beta_{21}k_1 + \beta_{22}k_2),$$
$$y_{n+1} = y_n + \gamma_1 k_1 + \gamma_2 k_2.$$

用 Taylor 级数展开,得到

$$k_q = hf + hf_i(\beta_{q1}k_1^i + \beta_{q2}k_2^i)$$

$$+\frac{h}{2}f_{ij}(\beta_{q1}k_1^i+\beta_{q2}k_2^i)(\beta_{q1}k_1^j+\beta_{q2}k_2^j)+\cdots.$$

将其代入右端 k_q 多次，得到

$$\begin{aligned}k_q&=hf+hf_i[\beta_{q1}(hf^i+hf_j^i(\beta_{11}k_1^j+\beta_{12}k_2^j)+\cdots)\\&\quad+\beta_{q2}(hf^i+hf_j^i(\beta_{21}k_1^j+\beta_{22}k_2^j)+\cdots)]\\&\quad+\frac{h}{2}f_{ij}(\beta_{q1}+\beta_{q2})hf^i(\beta_{q1}+\beta_{q2})hf^j+\cdots\\&=hf+h^2f_if^i\alpha_q+\frac{h^3}{2}f_{ij}f^if^j\alpha_q^2+h^3f_if_j^if^j(\beta_{q1}\alpha_1+\beta_{q2}\alpha_2)\\&\quad+\frac{h^4}{6}\alpha_q^3f_{ijk}f^if^jf^k+h^4f_{ij}f_k^if^jf^k\alpha_q(\beta_{q1}\alpha_1+\beta_{q2}\alpha_2)\\&\quad+\frac{h^4}{2}f_if_{jk}^if^jf^k(\beta_{q1}\alpha_1^2+\beta_{q2}\alpha_2^2)\\&\quad+h^4f_if_j^if_k^jf^k(\beta_{q1}(\beta_{11}\alpha_1+\beta_{12}\alpha_2)\\&\quad+\beta_{q2}(\beta_{21}\alpha_1+\beta_{22}\alpha_2))+O(h^5),\end{aligned}\tag{2.24}$$

其中 $\alpha_1=\beta_{11}+\beta_{12}$，$\alpha_2=\beta_{21}+\beta_{22}$. 计算 y_{n+1}，得到

$$\begin{aligned}y_{n+1}&=y_n+(\gamma_1+\gamma_2)hf+(\gamma_1\alpha_1+\gamma_2\alpha_2)h^2f_if^i\\&\quad+(\gamma_1\alpha_1^2+\gamma_2\alpha_2^2)\frac{h^3}{2}f_{ij}f^if^j+(\gamma_1(\beta_{11}\alpha_1+\beta_{12}\alpha_2)\\&\quad+\gamma_2(\beta_{21}\alpha_1+\beta_{22}\alpha_2))h^3f_if_j^if^j+O(h^4)\end{aligned}\tag{2.25}$$

从此通过与 Taylor 级数直到阶为 h^3 的项的一致，我们得到方程

$$\gamma_1+\gamma_2=1,$$
$$\gamma_1\alpha_1+\gamma_2\alpha_2=\frac{1}{2},$$
$$\gamma_1\alpha_1^2+\gamma_2\alpha_2^2=\frac{1}{3},$$
$$\gamma_1(\beta_{11}\alpha_1+\beta_{12}\alpha_2)+\gamma_2(\beta_{21}\alpha_1+\beta_{22}\alpha_2)=\frac{1}{6}.$$

这六个未知数四个方程的方程组有一个双参数解族. 从前面

三个方程,我们得到

$$\gamma_2(\alpha_2-\alpha_1)=\frac{1}{2}-\alpha_1,$$

$$\gamma_2(\alpha_2-\alpha_1)(\alpha_2+\alpha_1)=\frac{1}{3}-\alpha_1^2$$

或

$$\alpha_2=-\alpha_1+\frac{\frac{1}{3}-\alpha_1^2}{\frac{1}{2}-\alpha_1}.$$

于是,选择 $\alpha_1\neq\frac{1}{2}$ 来决定 α_2,因此,由前面三个方程决定 γ_1 和 γ_2. 因为 $\beta_{11}=\alpha_1-\beta_{12}$, $\beta_{21}=\alpha_2-\beta_{22}$, 所以,最后的方程是 β_{12} 和 β_{22} 的非线性方程. 选择其中一个来决定另一个,于是,一组解是

$$\alpha_1=0,\quad \alpha_2=\frac{2}{3},\quad \gamma_2=\frac{3}{4},\quad \gamma_1=\frac{1}{4},$$

$$\beta_{12}=1,\quad \beta_{22}=0,\quad \beta_{11}=-1,\quad \beta_{21}=\frac{2}{3},$$

得到

$$k_1=hf(y_n-k_1+k_2),$$

$$k_2=hf\left(y_2+\frac{2}{3}k_2\right),$$

$$y_{n+1}=y_n+\frac{1}{4}(k_1+3k_2).$$

这是一个三阶方法. 但是, Butcher (1964) 已经研究了 r 级隐式 Runge-Kutta 方法,并且已证明对于这样的方法[1] 达到 $2r$ 阶是可能的. 因此,如果在上述二阶方法中选取

1) Ceschino 和 Kuntzman (1966) 曾报告过这些方法,并得到这结果的一个证明.

$$\gamma_1 = \gamma_2 = \frac{1}{2},$$

$$\alpha_1 = \frac{1}{2} - \frac{\sqrt{3}}{6},$$

$$\alpha_2 = \frac{1}{2} + \frac{\sqrt{3}}{6},$$

$$\beta_{11} = \beta_{22} = \frac{1}{4},$$

$$\beta_{12} = \frac{1}{4} - \frac{\sqrt{3}}{6},$$

$$\beta_{21} = \frac{1}{4} + \frac{\sqrt{3}}{6},$$

那么，可以看到 h^4 的项与对应的 y_{n+1} 的 Taylor 级数的那些项一致．在 (2.25) 中，h^4 的项就是

$$\begin{aligned}
&\frac{(\gamma_1\alpha_1^3 + \gamma_2\alpha_2^3)f_{ijk}f^if^jf^k}{6} + [\gamma_1\alpha_1(\beta_{11}\alpha_1 + \beta_{12}\alpha_2) \\
&+ \gamma_2\alpha_2(\beta_{21}\alpha_1 + \beta_{22}\alpha_2)]f_{ij}f^i_kf^jf^k \\
&+ \frac{[\gamma_1(\beta_{11}\alpha_1^2 + \beta_{12}\alpha_2^2) + \gamma_2(\beta_{21}\alpha_1^2 + \beta_{22}\alpha_2^2)]f_if^i_{jk}f^jf^k}{2} \\
&+ [(\gamma_1\beta_{11} + \gamma_2\beta_{21})(\beta_{11}\alpha_1 + \beta_{12}\alpha_2) \\
&+ (\gamma_1\beta_{21} + \gamma_2\beta_{22})(\beta_{21}\alpha_1 + \beta_{22}\alpha_2)]f_if^i_jf^j_kf^k \\
&= \frac{[f_{ijk}f^if^jf^k + 3f_{ij}f^i_kf^jf^k + f_if^i_{jk}f^jf^k + f_if^i_jf^j_kf^k]}{24} \\
&= \frac{y^{(4)}}{24}.
\end{aligned}$$

2.5.1. 隐式 Runge-Kutta 方法的实际应用

显式 Runge-Kutta 方法使用起来简单明了，而隐式 Runge-

Kutta 方法都需要每步求解一个联立方程组. 如果 f 不是 y 的线性函数，这是非线性方程组. 它们可用对充分小的 h 保证收敛的迭代方法来求解：假定有 k_q 值的近似 $k_{q,(0)}, q=1, 2, \cdots, r$，用

$$k_{q,(m+1)} = hf(y_n + \beta_{qj}k_{j,(m)}) \tag{2.26}$$

确定新的近似值 $k_{q,(m+1)}$, $m=0, 1, 2, \cdots$. 假定在 k_q 的近似值中误差用

$$e_{q,(m)} = k_q - k_{q,(m)}$$

定义,那么,由(2.23)减去(2.26)得到

$$\begin{aligned}|e_{q,(m+1)}| &= |hf(y_n+\beta_{qj}k_j) - hf(y_n+\beta_{qj}k_{j,(m)})| \\ &\leqslant hL|\beta_{qj}(k_j - k_{j,(m)})| \text{（根据 Lipschitz 条件）} \\ &\leqslant hL|\beta_{qj}|\,|e_{j,(m)}|.\end{aligned}$$

于是，若 $e_{(m)} = \max\limits_q |e_{q,(m)}|$，则得

$$e_{(m+1)} \leqslant hL \max_q \sum_{j=1}^{r} |\beta_{qj}| e_{(m)}$$

并证明了：若

$$h < \frac{1}{L \max\limits_q \sum\limits_j |\beta_{qj}|},$$

则当 $m \to \infty$ 时，$e_{(m)} \to 0$.

一般来说，用已经达到的增加了的精确度来说明隐式方法的附加工作量的合理性是困难的,所以,它们的实用性仅限于一些特殊问题. 对这些问题来说，方法具有所需要的稳定性.

2.6. 收敛性和稳定性

我们将在第 4 章证明 Runge-Kutta 方法和 Taylor 级数方法,这些方法当$h \to 0$时阶大于零则收敛,还要证明收敛的单步方法是稳定的(即给问题一个小的扰动,当 $h \to 0$ 时只引起一

个有界的变化）.本节讨论 Runge-Kutta 方法的绝对稳定区域.

2.6.1. 显式 Runge-Kutta 方法的稳定区域

对于复值 λ 考虑方程 $y' = \lambda y$. 若对 $y' = \lambda y$ 研究四阶显式 Runge-Kutta 方法，则得到

$$k_1 = \lambda h(y + \alpha_1 h\lambda y) = h\lambda(1 + \alpha_1 h\lambda)y,$$

$$k_2 = \lambda h(1 + \beta_{21}h\lambda + \beta_{12}h\lambda(1 + \alpha_1 h\lambda))y$$

$$= h\lambda(1 + \alpha_2 h\lambda + \alpha_1\beta_{12}h^2\lambda^2)y.$$

类似地，

$$k_3 = h\lambda(1 + \eta_1 h\lambda + \eta_2 h^2\lambda^2 + \eta_3 h^3\lambda^3)y$$

并且

$$y_{n+1} = (1 + \delta_1 h\lambda + \delta_2 h^2\lambda^2 + \delta_3 h^3\lambda^3 + \delta_4 h^4\lambda^4)y_n,$$

其中 η_i 和 δ_i 都是系数 β_{ij} 和 α_j 的组合. 由于我们知道，如果 y_n 精确，则 y_{n+1} 与 $y(t_{n+1})$ 一致到 h^4 阶；又因为 $y(t_{n+1}) = e^{\lambda h}y(t_n)$, 所以，有 $\delta_1 = 1, \delta_2 = \frac{1}{2}, \delta_3 = \frac{1}{6}$ 和 $\delta_4 = \frac{1}{24}$. 于是，方法的放大系数是

$$\sum_{n=0}^{4}\frac{(h\lambda)^n}{n!}.$$

因此，绝对稳定区域是不等式

$$\left|1 + \mu + \frac{\mu^2}{2} + \frac{\mu^3}{6} + \frac{\mu^4}{24}\right| < 1 \qquad (2.27)$$

满足的区域，其中 $\mu = h\lambda$. 为了在复平面内找出这个区域，我们画出使 (2.27) 等号成立的轨迹. 得到这曲线的一个办法是令

$$1 + \mu + \frac{\mu^2}{2} + \frac{\mu^3}{6} + \frac{\mu^4}{24} = e^{i\theta},$$

其中 $\mu = h\lambda$, 并且用标准的多项式求根法确定 $\mu(\theta)$. 绝对稳

定区域在图 2.1 中表出.

如果考虑具有解

$$y = F(t) + c_0 e^{\lambda t}$$

的问题

$$y' = \lambda(y - F(t)) + F'(t), \ y(0) = F(0) + c_0.$$

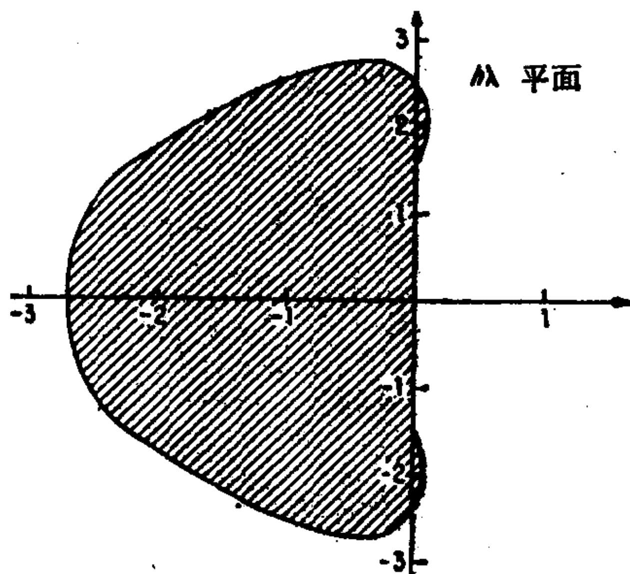


图 2.1. 四阶显式 Runge-Kutta 方法的绝对稳定区域

我们注意到,因为它对 y 是线性的,所以,可以把数值解看做是分别积分非齐次问题

$$y' = \lambda(y - F(t)) + F'(t), \ y(0) = F(0) \quad (2.28)$$

和齐次问题

$$y' = \lambda y, \ y(0) = c_0 \quad (2.29)$$

的结果之和. 为了能精确地积分 (2.28) 的 $F(t)$项, 步长无疑要取得足够小.我们假定 $F(t)$ 是光滑的且量级保持在 1 左右. 如果 $\lambda \ll 0$ (或者它的实部 $\ll 0$), 那么, 开始步长一定很小致使能够精确积分 $c_0 e^{\lambda t}$ 项. 因为 $e^{h\lambda}$ 用 $1 + h\lambda + [(h\lambda)^2/2] + [(h\lambda)^3/6] + [(h\lambda)^4/24]$ 来数值表示,所以, $h\lambda$ 也必须足够小, 使得这个表达式接近于 $e^{h\lambda}$. 许多步之后, 与解中

$F(t)$ 项比较，$C_0e^{\lambda t}$ 项就不那么重要了．从此，绝对稳定性就是步长的限制性需要．h 一定得充分小，使得 $h\lambda$ 在绝对稳定的区域之中，以便对 $e^{\lambda t}$ 项的数值近似不增加．然而，这个 h 比精确地积分 $F(t)$ 项所必需的步长要小得多．当 λ 很小且为负时，为了准确地积分 $F(t)$ 所选取的 h 值使得 $h\lambda$ 在绝对稳定区域内．当 $\lambda > 0$ 时，(2.29) 的解增长，且 h 足够小，使得也可正确地积分 $e^{\lambda t}$．但是，$h\lambda$ 不落在绝对稳定区域中，这点不是重要的，因为 $e^{\lambda t}$ 也在增长．

对于$\lambda < 0$，或者考虑精确度，或者注意绝对稳定性（前面要求限制$h\lambda$，使其对于大多数方法是在绝对稳定区域之内，因为如果 $1 + h\lambda + \cdots$ 是 $e^{\lambda h}$ 的一个准确近似值，那么，当 $h\lambda < 0$ 时，一定小于 1）．如果 $\lambda > 0$，那么，精确度是唯一的准则．

q 级显式 Runge-Kutta 方法用 $1 + h\lambda + \cdots + (h\lambda)^q/q!$ 表示 $e^{h\lambda}$，因此，如果方法是 q 阶的（如果 $q \leqslant 4$ 是可能的），那么，表示式就是指数级数的前 $(q + 1)$ 项．用到 q 阶导数的 Taylor 级数方法将有类似的性质．

2.6.2. 隐式 Runge-Kutta方法的稳定区域

虽然隐式方法在实用方面并不十分重要，但是它有一个重要的稳定特性，使其对于后面有待讨论的若干特殊问题是有用的．本节用简单例子（如梯形法则）来说明这种性质．一般 r 级隐式 Runge-Kutta 方法的推广可在文献中找到．

梯形法则由

$$y_{n+1} = y_n + \frac{h}{2}(y'_n + y'_{n+1})$$

给出．若代入 $y' = \lambda y$，则得到

$$y_{n+1} = y_n + \frac{h\lambda}{2}y_n + \frac{h\lambda}{2}y_{n+1}$$

或

$$y_{n+1}=\frac{1+(h\lambda/2)}{1-(h\lambda/2)}y_n.$$

于是，增长因子是 $[1+(h\lambda/2)]/[1-(h\lambda/2)]$. 如果 $h\lambda/2=u+iv$，其中 u 和 v 是实的，则得

$$|y_{n+1}|=\left|\frac{(1+u)+iv}{(1-u)-iv}\right||y_n|$$
$$=\left|\frac{1+u^2+v^2+2u}{1+u^2+v^2-2u}\right|^{1/3}|y_n|.$$

所以，若 $u<0$，解是衰减的. Dahlquist (1963) 定义一个方法是 A-稳定的，指的是：对于微分方程 $y'=\lambda y$，其中 $\mathrm{Re}(\lambda)<0$，当 $n\to\infty$ 时数值解渐近逼近于零. 梯形法则是 A-稳定的.

如 Ehle (1968) 所指出，$2r$ 阶的 r 级隐式 Runge-Kutta 方法也是 A-稳定的. 例如，用步长 $h=1$, 0.1 和 0.001 及梯形方法来考虑问题

$$y'=-1000(y-t^3)+3t^2,\quad y(0)=0,\quad y(1)=?$$

解列在表 2.6 中. 注意，误差按 $O(h^2)$ 变化，不管 $h\left(\frac{\partial f}{\partial y}\right)$ 开头的庞大值是 -1000.

表 2.6. 梯形方法（作为 h 的函数）的误差

h	$y_h(1)$	$\epsilon_h(1)=y_h(1)-1$	$\epsilon_h(1)/h^2$
1	1.00998000	0.99800×10^{-3}	0.998×10^{-3}
0.1	1.00000165	0.16486×10^{-5}	0.165×10^{-3}
0.01	1.00000005	0.50000×10^{-7}	0.500×10^{-3}
0.001	1.00000000	0.49996×10^{-9}	0.500×10^{-3}

问　　题

1. 如果采用下列方法并达到要求的精度，积分方程 $y'=y$, $y(0)=1$，计算从 $t=0$ 到 $t=1$ 应该使用的固定步长的最小步数.

(a) 改进的梯形方法，至少具有精度 0.20;0.05.

(b) 经典的四阶 Runge-Kutta 方法，至少具有精度 0.01; 0.001.

2. 用方程 (2.8) 给出的改进的梯形法则及方程 (2.7) 给出的通常的梯形法则，在区间 [0,1] 上用步长 2^{-p}, $p = 2(2)12$ (即 $p = 2, 4, 6, 8, 10, 12$) 积分

$$y' = -1000(y - t^3) + 3t^2, \ y(0) = 0,$$

并在对数座标纸上画出这个结果的图形. 说明它们的共性和差异.

3. 欲积分一般方程

$$y' = f(y, t),$$

从上个问题的结果看，梯形法则比改进的梯形法则效果好吗?

4. 假定用中点法则积分

$$y' = \lambda(t^3 - y) + 3t^2, \ y(0) = 0,$$

从 $t = 0$ 到 $t = 1$，误差为 $O(h^3)$，那么，λ 应取什么样的值. 利用对所求 λ 值数值积分并画出 h 的误差曲线来说明这一点.

5. 从一般四阶 Runge-Kutta 公式导出误差项 $O(h^5)$，并对经典四阶 Runge-Kutta 方法求其误差项的特殊形式. 若假定 f 对于 y 和 t 是线性的，那么，一般公式的值如何?

6. 计算并画出三阶 Runge-Kutta 方法的绝对稳定区域.

7. 讨论由

$$y_{n+1} = y_n + \frac{h}{2}(f(y_n) + f(y_{n+1})) + \frac{h^2}{12}(f'(y_n) - f(y_{n+1}))$$

定义的方法的绝对稳定区域，其中 $f'(y)$ 是由 $f'(y) = (\partial f/\partial t) + (\partial f/\partial y)f(y)$ 计算的，这个方法的阶是多少?

3. 方程组和高阶方程

这一章的目的是将前两章的方法推广到两种情形，即推广到方程组和高阶方程．所谓方程的阶是方程中出现的导数的最高阶数．假定方程形如

$$y^{(p)} = f(y, y', \cdots, y^{(p-1)}, t), \tag{3.1}$$

其中 f 对每个 $y^{(i)}$ 满足 Lipschitz 条件，而 $y^{(i)}$ 是 d^iy/dt^i 的一种记号．于是

$$(y''')^2(2 + t^2) - y(y' + \cos t) = 0$$

就是一个三阶方程，又可写成

$$y''' = \left[\frac{y(y' + \cos t)}{2 + t^2}\right]^{1/2}. \tag{3.2}$$

但是，必须要求 $y(y' + \cos t)$ 保持离开零为有界，以便(3.2)满足 Lipschitz 条件，而且必须取两平方根之一，使得 y''' 为单值．

方程组包括一个自变量 t 和多个因变量．于是，

$$\begin{cases} y' = z, \\ z' = -y \end{cases} \tag{3.3}$$

是两个一阶方程的方程组．

如果因变量有 s 个，则记作 $y^1, y^2, y^3, \cdots, y^s$．上标不会与 y 的方幂混淆，如果有可能混淆时，y 的方幂记作 $(y)^p$．一般的一阶方程组形如

$$\begin{cases} y^{1'} = f^1(y^1, y^2, \cdots, y^s, t), \\ y^{2'} = f^2(y^1, y^2, \cdots, y^s, t), \\ \cdots\cdots \\ y^{s'} = f^s(y^1, y^2, \cdots, y^s, t), \end{cases} \tag{3.4}$$

假定其中每个 f^i 分别对各个因变量满足 Lipschitz 条件且对 t 连续．如果像第 2 章做过的一样重新取 t 为 y^0，而且加上初始条件为 $y^0(0)=1$ 的微分方程

$$y^{0\prime}=f^0(y^0, y^1, \cdots, y^s)=1, \qquad (3.5)$$

则以后的符号就被简化了．方程 (3.4) 和 (3.5) 现在是自变量为 t 的 $s+1$ 个因变量 $y^0,\cdots,y^s$ 的 $s+1$ 个方程的方程组，于是可写成

$$\mathbf{y}'=\mathbf{f}(\mathbf{y}), \qquad (3.6)$$

其中 $\mathbf{y}$ 是列向量 $[y^0, y^1, \cdots, y^s]^T$，$T$ 是转置算子．关于这种改写有两点应当强调：首先，如果要求 f^i 对 y^0 满足 Lipschitz条件，则需有一个比对 t 连续还要强一些的条件，而对 t 连续是所有方法的存在性和收敛性的充分条件．但是，实际上甚至有更强的可微性质，所以这不成问题．其次，这个变换的目的只是为了统一符号，为了减少工作量和舍入误差，在大多数数值积分中，分别处理自变量．如果步长是固定的，那么最好是由 h_n 计算 t_n．但是，如果步长 h 不固定，那么 t_n 也许不得不由 $t_{n+1}=t_n+h_n$ 来计算，如果 h 在机器中不是精确的表示，则肯定会引进一些舍入误差，我们必须承认这种事实．

3.1. 单步方法应用于方程组

第 1 章和第 2 章所讨论的全部方法均可直接应用于方程组．方程组中任一方程既可分别讨论又可联立在一起，于是，对于

$$\begin{aligned} y' &= f(y, z) \\ z' &= g(y, z), \end{aligned} \qquad (3.7)$$

Euler 方法是

$$y_{n+1}=y_n+hf(y_n, z_n),$$

$$z_{n+1} = z_n + hg(y_n, z_n).$$

经典的四级 Runge-Kutta 方法要求对每个因变量计算一组 k_q，于是，如果把(3.7)中的 y 记作 y^1，z 记作 y^2，则可应用由

$$
\begin{aligned}
k_0^1 &= hf(y_n^1, y_n^2),\\
k_0^2 &= hg(y_n^1, y_n^2),\\
k_1^1 &= hf\left(y_n^1 + \frac{1}{2}k_0^1, y_n^2 + \frac{1}{2}k_0^2\right),\\
k_1^2 &= hg\left(y_n^1 + \frac{1}{2}k_0^1, y_n^2 + \frac{1}{2}k_0^2\right),\\
k_2^1 &= hf\left(y_n^1 + \frac{1}{2}k_1^1, y_n^2 + \frac{1}{2}k_1^2\right),\\
k_2^2 &= hg\left(y_n^1 + \frac{1}{2}k_1^1, y_n^2 + \frac{1}{2}k_1^2\right),\\
k_3^1 &= hf(y_n^1 + k_2^1, y_n^2 + k_2^2),\\
k_3^2 &= hg(y_n^1 + k_2^1, y_n^2 + k_2^2),\\
y_{n+1}^1 &= y_n^1 + \frac{1}{6}(k_0^1 + 2k_1^1 + 2k_2^1 + k_3^1),\\
y_{n+1}^2 &= y_n^2 + \frac{1}{6}(k_0^2 + 2k_1^2 + 2k_2^2 + k_3^2).
\end{aligned}
\tag{3.8}
$$

给出的 Runge-Kutta 方法．第 4 章将证明；这些方法具有适于单个方程的类似方法的同样性质．我们注意，第 2 章中有关 Taylor 级数展开式的分析，可以直接应用．事实上，那里为了处理自变量所导出的符号与这里使用的相同，所以，由方程(2.13)到(2.15)给出的导数的展开式对方程组亦成立，其中 i，j 等不仅可以取值 0 和 1，而且允许取值 0 到 s．

3.2. 高阶方程简化为一阶方程组

为了处理形如(3.1)的方程，通常的方法是把它变换成等价的一阶方程组．如果定义变量

$$y^i = y^{(i-1)},\ i = 1, 2, \cdots, p, \tag{3.9}$$

则可把 (3.1) 写成

$$(y^p)' = f(y^1, y^2, \cdots, y^p, t). \tag{3.10a}$$

通过 (3.9) 对 $i = 1, 2, \cdots, p-1$ 的微分,我们得到

$$(y^i)' = y^{(i)} = y^{i+1}. \tag{3.10b}$$

(3.10) 是 p 个一阶方程组,它可用已经讨论过的方法来处理. 原来的问题 (3.1) 具有由 $y^{(i-1)}(0)(i = 1, 2, \cdots, p)$ 确定的初始值, 现在的 (3.10) 只要具有 $y^i(0)$ 所需要的确定的初始值.

3.3. 高阶方程的直接方法

许多人提出了直接方法, 而不去把一个高阶方程展成庞大的方程组,其中一些方法将在后面多步方法的章节中讨论. 这里略述单步方法推广到高阶方程,例如 (3.1). 是直接处理这些方程较好,还是把这些方程变成低阶方程组较好,依赖于方程本身. 某些情况已经在 Rutishauser (1960) 的论文中研究过了.

3.3.1. Taylor 级数方法

Taylor 级数方法可用显然的方式推广如下: 给定 y_0, $y_0^{(1)}, \cdots, y_0^{(p-1)}$,利用 (3.1) 能够计算 $y_0^{(p)}$,并且通过微分(3.1)多次,还能得到 $y_0^{(q)}$, $q = p+1, \cdots, r$. 这样,可书写如下:

$$y_1 = y_0 + hy_0^{(1)} + \frac{h^2}{2}y_0^{(2)} + \cdots + \frac{h^r}{r!}y_0^{(r)},$$

$$y_1^{(1)} = y_0^{(1)} + hy_0^{(2)} + \frac{h^2}{2}y_0^{(3)} + \cdots + \frac{h^{r-1}}{(r-1)!}y_0^{(r)},$$

$$\cdots$$

$$y_1^{(p-1)} = y_0^{(p-1)} + hy_0^{(p)} + \cdots + \frac{h^{r-p+1}}{(r-p+1)!}y_0^{(r)}. \tag{3.11}$$

因此，y，$y^{(1)}$，$\cdots$，$y^{(p-1)}$ 的值能够在 t_1 点近似算出。

3.3.2. Runge-Kutta 方法

像在一阶方程的 Runge-Kutta 方法中做过的一样，不直接计算比 p 更高阶的导数，只在多个点上计算阶为 p 的导数值．例如，考虑二阶方程 $y''=f(y,y')$．已知 y_n，y'_n，一般的二级显式 Runge-Kutta 方法形如

$$
\begin{aligned}
k_1 &= \frac{h^2}{2}f(y_n,\ y'_n),\\
k_2 &= \frac{h^2}{2}f\left(y_n+\alpha_1 hy'_n+\alpha_2 k_1,\ y'_n+\frac{\alpha_3 k_1}{h}\right),\\
y_{n+1} &= y_n+hy'_n+\gamma_1 k_1+\gamma_2 k_2,\\
y'_{n+1} &= y'_n+\frac{\delta_1}{h}k_1+\frac{\delta_2}{h}k_2.
\end{aligned}
\tag{3.12}
$$

展开 k_2，得到

$$
\begin{aligned}
k_2 = \frac{h^2}{2}\Bigg[&f+(\alpha_1 hy'+\alpha_2 k_1)\frac{\partial f}{\partial y}+\alpha_1^2\frac{h^2(y')^2}{2}\frac{\partial^2 f}{\partial y^2}\\
&+\frac{\alpha_3 k_1}{h}\frac{\partial f}{\partial y'}+\frac{\alpha_3^2 k_1^2}{2h^2}\frac{\partial^2 f}{\partial y'^2}+\alpha_1\alpha_3 k_1\frac{\partial^2 f}{\partial y\partial y'}y'\Bigg]+O(h^5),
\end{aligned}
\tag{3.13}
$$

其中每一个量在 y_n，y'_n 上计算．我们还有 $y''_n=f$，$y'''_n=(\partial f/\partial y)y'+(\partial f/\partial y')y''$，

$$
\begin{aligned}
y_n^{(4)} = &\frac{\partial^2 f}{\partial y^2}(y')^2+2\frac{\partial^2 f}{\partial y\partial y'}y'y''+\frac{\partial f}{\partial y}y''\\
&+\frac{\partial^2 f}{\partial y'^2}(y'')^2+\frac{\partial f}{\partial y'}y'''.
\end{aligned}
$$

把 (3.13) 代入 (3.12)，得到

$$
y_{n+1}=y_n+hy'_n+(\gamma_1+\gamma_2)\frac{h^2}{2}y''_n
$$

$$+\gamma_2\frac{h^3}{2}\left[\alpha_1\frac{\partial f}{\partial y}y'+\frac{\alpha_3}{2}\frac{\partial f}{\partial y'}y''\right]$$
$$+\gamma_2\frac{h^4}{4}\left[\alpha_2\frac{\partial f}{\partial y}y''+\alpha_1^2\frac{\partial^2 f}{\partial y^2}(y')^2\right.$$
$$\left.+\alpha_1\alpha_3\frac{\partial^2 f}{\partial y\partial y'}y'y''+\frac{\alpha_3^2}{4}\frac{\partial^2 f}{\partial y'^2}(y'')^2\right]+O(h^5). \quad (3.14)$$

如果 $\gamma_1+\gamma_2=1$ 且 $\alpha_1\gamma_2=\alpha_3\gamma_2/2=\frac{1}{3}$，那么，(3.14) 与 $y(t_{n+1})$ 的 Taylor 展式前四项一致．h^4 项对于 α_i 和 γ_i 的任何选取都不与 Taylor 级数的第五项一致．类似地，如果 $\delta_1+\delta_2=2$ 以及 $\alpha_1\delta_2=\alpha_3\delta_2/1=1$，则 y'_{n+1} 与 $y'_n+hy''_n+(h^2/2)y'''_n$ 一致到阶 h^2．于是，一组解是

$$\gamma_1=\gamma_2=\frac{1}{2},$$
$$\alpha_1=\alpha_2=\frac{2}{3},\qquad \alpha_3=\frac{4}{3},$$
$$\delta_1=\frac{1}{2},$$
$$\delta_2=\frac{3}{2}. \quad (3.15)$$

局部截断误差对 y 是 $O(h^4)$，而对 y' 是 $O(h^3)$．因此，可以假定全体误差分别是 $O(h^3)$ 和 $O(h^2)$．但是，y' 影响到 f，因而影响到 k_2，所以，对于 y 和 y'，全体误差均为 $O(h^2)$．

对于高阶方程组，已得到显式[1] Runge-Kutta 方法和隐式[2] Runge-Kutta 方法．这些方法在若干特殊问题中有着重要的应用，但是，一般的计算机程序通常只是对一阶方程编制的．如果方程是特殊的类型，则这些方法会更实用的．缺少一阶导

1) Zurmuhl (1968)和 Henrici (1962)，见 4.2.3.
2) Cooper (1967).

数的方程组

$$\mathbf{y}'' = \mathbf{f}(\mathbf{y})$$

在天体力学中常常出现．在简单的情形，$y'' = f(y)$，$y^{(3)} = f_y y'$，$y^{(4)} = f_y y'' + f_{yy}(y')^2$ 及 $f_{y'} = 0$．因此，(3.14) 由

$$y_{n+1} = y_n + hy_n' + \frac{h^2}{2} y_n''(\gamma_1 + \gamma_2) + \frac{h^3}{2}\gamma_2\alpha_1 f_y y' + \frac{h^4}{4}\gamma_2[\alpha_2 f_y y'' + \alpha_1^2 f_{yy}(y')^2] + O(h^5) \tag{3.16}$$

所代替，而 y_{n+1}' 由

$$y_{n+1}' = y_n' + \frac{h}{2} y_n''(\delta_1 + \delta_2) + \frac{h^2}{2}\delta_2\alpha_1 f_y y' + \frac{h^3}{4}\delta_2[\alpha_2 f_y y'' + \alpha_1^2 f_{yy}(y')^2] + O(h^4) \tag{3.17}$$

给出．通过选取

$$\delta_2\alpha_2 = \frac{2}{3}, \qquad \delta_2\alpha_1^2 = \frac{2}{3},$$

$$\delta_1 + \delta_2 = 2, \qquad \delta_2\alpha_1 = 1,$$

$$\gamma_2\alpha_1 = \frac{1}{3}, \qquad \gamma_1 + \gamma_2 = 1,$$

(3.16) 和 (3.17) 的前四项与 Taylor 级数的对应项一致．它给出了特殊二阶方程的 Nyström 公式：

$$k_1 = h^2 f(y_n, t_n)/2,$$

$$k_2 = \frac{h^2 f[y_n + (2hy_n'/3) + (4k_1/9)]}{2},$$

$$y_{n+1} = y_n + hy_n' + \frac{1}{2}(k_1 + k_2),$$

$$y_{n+1}' = y_n' + \frac{k_1 + 3k_2}{2h}.$$

这个公式对 y 和 y' 都具有 $O(h^4)$ 的局部截断误差，对每个分

量的全体误差为 $O(h^3)$.

问　　题

1. 对于二阶方程

$$y'' = f(y, y', t),$$

考虑形如

$$y'_{n+1} = y'_n + \alpha_1 h f_n + \alpha_2 h f_{n+1},$$
$$y_{n+1} = y_n + h y'_n + \beta_1 h^2 f_n + \beta_2 h^2 f_{n+1}$$

的方法,这里选取的 f_{n+1} 指的是 $f(y_{n+1}, y'_{n+1}, t_{n+1})$. 在全体误差上能够达到的最大阶数是多少? 若 $\partial f/\partial y' \equiv 0$, 阶数改变吗? 给出达到你所得到的阶的方法的例子的系数.

2. 证明方法

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \frac{1}{2} h [\mathbf{f}(\mathbf{y}_n) + \mathbf{f}(\mathbf{y}_{n+1})] + \frac{1}{12} h^2 [\mathbf{f}'(\mathbf{y}_n) - \mathbf{f}'(\mathbf{y}_{n+1})]$$

是对一阶方程组的四阶方法.

3. 对于特殊的三阶方程

$$y''' = f(y),$$

你能找到的二级 Runge-Kutta 型最高阶方法是什么样? 而二级方法形如

$$k_0 = \frac{h^3 f(y_n)}{6},$$

$$k_1 = h^3 f\left(y_n + \alpha h y'_n + \alpha^2 \frac{h^2}{2} y''_n + \alpha^3 k_0\right),$$

$$y_{n+1} = y_n + h y'_n + \frac{h^2}{2} y''_n + \beta_1 k_0 + \beta_2 k_1,$$

$$y'_{n+1} = y'_n + h y''_n + \gamma_1 \frac{k_0}{h} + \gamma_2 \frac{k_1}{h},$$

$$y''_{n+1} = y''_n + \delta_1 \frac{k_0}{h^2} + \delta_2 \frac{k_1}{h^2}.$$

4. 在你得出的问题 3 的答案中, 描述的二级 Runge-Kutta 方法的最高阶能继续保持,更一般的三阶方程的形式是什么?

4. 单步方法的收敛性、误差界和误差估计

这一章研究如下问题:

1. 在什么条件下方法收敛?
2. 收敛的阶是多少?
3. 可以获得何种类型的误差界?
4. 当 $h \to 0$ 时,关于误差的渐近形式可以讨论什么内容(如定理 1.4 对 Euler 方法的处理)?

第 1 章研究了最简单的单步方法，现在研究下面的单步方法:

1. 对于一阶单个方程的高阶方法;
2. 对于一阶方程组的方法;
3. 对于阶$\geqslant 1$的方程组的方法.

每一类是其后一类的特殊情况，所以，我们证明(3)类的结果并讨论将这些定理应用到其他类问题的较简单情形上去.

虽然我们将讨论高阶方程组的直接方法,但是,几乎总是应用 3.2 中记法的简化来讨论更大的一阶方程组.例如，有一对方程:

$$y'' = f(y, y', z, z', z'', t),$$
$$z''' = g(y, y', z, z', z'', t).$$

记

$$y^1 = y,\ y^2 = y',\ y^3 = z,\ y^4 = z',\ y^5 = z'',$$

我们可将其表示成方程组

$$\mathbf{y}' = \mathbf{f}(\mathbf{y}, t),$$

其中

$$f^1 = y^2,$$
$$f^2 = f(y^1, y^2, y^3, y^4, t),$$
$$f^3 = y^4,$$
$$f^4 = y^5,$$
$$f^5 = g(y^1, y^2, y^3, y^4, t).$$

4.1. 向量和矩阵模

当我们讨论单个方程 Euler 方法的误差时，给出了其绝对值的界．当我们处理方程组时，它的每一个元都会有误差，于是有一误差向量．求得误差界是必要的，它保证这些误差全是小的．这样，有一个单个的量来度量所有这些误差的大小是方便的．这个量应该具有尽可能多的绝对值运算的性质，使得有关单个方程的定理可以直接推广．我们称这样的量为向量 $\mathbf{a}$ 的模，记作 $\|\mathbf{a}\|$．模的一个例子是 $\|\mathbf{a}\| = \max_i |a^i|$，其中 a^i 是 $\mathbf{a}$ 的分量，称为最大模．如果 $\|\mathbf{a}\| \leqslant \varepsilon$，那么，每个 $|a^i| \leqslant \varepsilon$．因此，如果限制 $\|\mathbf{a}\|$，则限制了每个分量．可以定义许多种不同的模，在形式上它们必须满足这样的性质：$\|\mathbf{a}\|$ 是非负实数，使得

$$\|\mathbf{a}\| = 0 \iff \mathbf{a} = \mathbf{0} \tag{4.1}$$
（这就保证了当模收敛于零时，向量的分量也收敛于零），

$$\|\mathbf{a} + \mathbf{b}\| \leqslant \|\mathbf{a}\| + \|\mathbf{b}\| \tag{4.2}$$
（这是三角不等式，允许将复杂的表达式展开并将每一部分分别求界），

对于任何纯量 λ，

$$\|\lambda \mathbf{a}\| = |\lambda| \|\mathbf{a}\|. \tag{4.3}$$

一向量可以左乘一矩阵而获得另一向量．矩阵的模记作 $\|A\|$．如果对于所有矩阵 A 和向量 $\mathbf{a}$，有

$$\|A\mathbf{a}\| \leqslant \|A\| \|\mathbf{a}\|, \tag{4.4}$$

则称矩阵模与向量模相容。矩阵模还必须满足性质(4.1),(4.2),(4.3)和

$$\|AB\| \leqslant \|A\|\|B\|. \tag{4.5}$$

可以看出，若 $\|A\|$ 由 $\sup\limits_{\|a\|\neq 0} \|A\mathbf{a}\|/\|\mathbf{a}\|$ 定义，则其满足(4.1)到(4.4). 上面给出的最大模的相容矩阵模是

$$\|A\| = \max_i \sum_i |A^i_j|.$$

注意到

$$\begin{aligned}\|A\mathbf{a}\| &= \max_i \left| \sum_j A^i_j a^j \right| \\ &\leqslant \max_i \sum_j |A^i_j| \mathop{\mathrm{mxa}}_k |a^k| \\ &= \|A\|\|\mathbf{a}\|,\end{aligned}$$

可以证明方程(4.4)，同时(4.5)由

$$\begin{aligned}\|AB\| &= \max_i \sum_k \left| \sum_j A^i_j B^j_k \right| \\ &\leqslant \max_i \sum_j \left[|A^i_j| \sum_k |B^j_k| \right] \\ &\leqslant \max_i \sum_j |A^i_j| \max_l \sum_k |B^l_k| \\ &= \|A\|\|B\|\end{aligned}$$

得出.

模的另一个例子是

$$\|\mathbf{a}\| = \sum_i |a^i|. \tag{4.6}$$

它有相容的矩阵模

$$\|A\| = \max_j \sum_i |A^i_j|. \tag{4.7}$$

我们经常引用 L_p 模. 所谓 L_p 模是指当 $p=1$ 时

$$\|\mathbf{a}\|_p = \left(\sum_i |a^i|^p \right)^{1/p}.$$

这是上面讨论过的 $\sum_i |a^i|$. 对于 $p=\infty$ (即当 $p\to\infty$ 时的极限), 我们得到最大模. $p=2$ 对应于熟知的距离 Euclid 模

$$\|a\|_2=\sqrt{\sum_i |a^i|^2}.$$

虽然最大模通常是最简单的，但以后几乎可以使用任何模. 不同的模可以导致不同的误差界，而最大模经常产生最明确的界.

4.2. 存在性和 Lipschitz 条件

定理 1.1 的直接模为任意阶方程组提供了一个存在性证明. 如果这些方程记为

$$\mathbf{y}'=\mathbf{f}(\mathbf{y},t)$$

我们可以叙述如下:

定理 4.1. 如果在区域 $0\leqslant t\leqslant b$, $-\infty\leqslant y\leqslant\infty$ 中, $\mathbf{f}(\mathbf{y},t)$是 t 的连续函数且关于 $\mathbf{y}$ 满足 Lipschitz 条件,那么存在唯一的可微函数 $\mathbf{y}(t)$, 使得

$$\mathbf{y}(0)=\mathbf{y}_0,$$

$$\frac{d\mathbf{y}(t)}{dt}=\mathbf{f}(\mathbf{y}(t),t).$$

这个结果在微分方程课本中可以找到.

所谓关于方程组的 Lipschitz 条件,意指存在一常数 L,使得

$$\|\mathbf{f}(t,\mathbf{y})-\mathbf{f}(t,\mathbf{y}^*)\|\leqslant L\|\mathbf{y}-\mathbf{y}^*\|. \tag{4.8}$$

如果 $\mathbf{f}$ 在这个意义下满足 Lipschitz 条件,那么它的每一个因变量也分别满足 Lipschitz 条件;反之亦然.

实际上，我们仅在 $\mathbf{y}$ 可以被证明属于其中的一个有限区域(根据 $\|f\|$ 的界)需要 Lipschitz 条件.如果在此区域上, $\mathbf{f}$ 有

对 $\mathbf{y}$ 的连续导数，那么它满足 Lipschitz 条件，因为由中值定理

$$f^i(t, \mathbf{y}) - f^i(t, \mathbf{y}^*) = \sum_j \frac{\partial f^i}{\partial y^j}(\xi^j)(y^j - y^{j*}),$$

其中 $\{\xi^j\}$ 是区域内的一组点. 如果

$$K_j^i = \max_{\text{区域}} \left| \frac{\partial f^i}{\partial y^j} \right|,$$

那么

$$|f^i(t, \mathbf{y}) - f^i(t, \mathbf{y}^*)| \leqslant \sum_j K_j^i |y^j - y^{j*}|.$$

如果定义 Lipschitz 常数 L 是矩阵 $K = \{K_j^i\}$的模，则得 (4.8).

4.3. 收敛性和稳定性

前面几章叙述的方法，其共同基础是规定一个量，加到 $\mathbf{y}_n$ 上，得到 $\mathbf{y}_{n+1}$. 形式上我们定义单步方法如下:

定义 4.1. 求微分方程近似解的单步方法，是一个可以写成如下形式:

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h\boldsymbol{\phi}(\mathbf{y}_n, t_n, h) \tag{4.9}$$

的方法，其中增量函数 $\boldsymbol{\phi}$ 是由 $\mathbf{f}$ 确定的，且仅仅是 $\mathbf{y}_n$, t_n 和 h 的函数.

单步方法的收敛性定义如下:

定义 4.2. 单步方法 (4.9) 是收敛的，如果对于任何满足 Lipschitz 条件的微分方程 $\mathbf{y}' = \mathbf{f}(\mathbf{y})$，当 $h = t/n$, $n \to \infty$ 和 $\mathbf{y}_0 \to \mathbf{y}(0)$ 时，对所有的 $0 \leqslant t \leqslant b$ 有 $y_n \to y(t)$.

我们注意到在初值 $\mathbf{y}_0$ 中允许有误差，因为实际上不能用有限的精度来精确表示 $\mathbf{y}(0)$. 收敛性使我们确信，使 h 很小和使用很高的精确度，能够任意充分逼近真解.

稳定性涉及到由数值解得来的扰动的影响.

定义 4.3. 单步方法是稳定的，如果对于每一个满足

Lipschitz 条件的微分方程，存在正常数 h_0 和 K，使得满足 (4.9) 的两个不同的数值解 $\mathbf{y}_n$ 和 $\tilde{\mathbf{y}}_n$ 之差对所有的 $0 \leqslant h \leqslant h_0$ 均有

$$\|\mathbf{y}_n - \tilde{\mathbf{y}}_n\| \leqslant K\|\mathbf{y}_0 - \tilde{\mathbf{y}}_0\|.$$

如下面定理所指出的，对于单步方法的稳定性，几乎是自动满足的.

定理 4.2. 如果 $\boldsymbol{\phi}(\mathbf{y}, t, h)$满足 Lipschitz 条件 L，则由 (4.9) 给出的方法是稳定的.

证明留下，作为一个简单的练习.

假设 $\mathbf{f}$ 始终满足定理 4.1 的条件，那么可以看出，对于前面讨论的所有方法，当 $0 \leqslant h \leqslant h_0$ 时，$\boldsymbol{\phi}$ 也满足这些条件.

例如，在中点法则的情况下，

$$\boldsymbol{\phi}(\mathbf{y}, t, h) = \mathbf{f}\left(\mathbf{y} + \frac{h}{2}\mathbf{f}(\mathbf{y}, t), t + \frac{h}{2}\right).$$

它关于 t 和 $\mathbf{y}$ 是连续的，如果 $\mathbf{f}$ 亦如此，并且

$$\begin{aligned}
&\|\boldsymbol{\phi}(\mathbf{y}, t, h) - \boldsymbol{\phi}(\mathbf{y}^*, t, h)\| \\
&= \left\| \mathbf{f}\left(\mathbf{y} + \frac{h}{2}\mathbf{f}(\mathbf{y}, t), t + \frac{h}{2}\right) \right. \\
&\qquad \left. - \mathbf{f}\left(\mathbf{y}^* + \frac{h}{2}\mathbf{f}(\mathbf{y}^*, t), t + \frac{h}{2}\right) \right\| \\
&\leqslant L \left\| \mathbf{y} + \frac{h}{2}\mathbf{f}(\mathbf{y}, t) - \mathbf{y}^* - \frac{h}{2}\mathbf{f}(\mathbf{y}^*, t) \right\| \\
&\leqslant L\|\mathbf{y} - \mathbf{y}^*\| + L\frac{h}{2}\|\mathbf{f}(\mathbf{y}, t) - \mathbf{f}(\mathbf{y}^*, t)\| \\
&\leqslant L\left(1 + \frac{Lh}{2}\right)\|\mathbf{y} - \mathbf{y}^*\|,
\end{aligned}$$

则对于 $0 \leqslant h \leqslant h_0$，$\boldsymbol{\phi}$ 关于 $\mathbf{y}$ 满足 Lipschitz 条件. 同时注意，如果 $\mathbf{f}$ 关于 $\mathbf{y}$ 和 t 是连续的，则 $\boldsymbol{\phi}$ 关于 h 是连续的. 在 Euler 方法情况下，$\boldsymbol{\phi} = \mathbf{f}(\mathbf{y}, t)$. 所以，我们能够证明下述定理，并

不意外.

定理 4.3. 如果对于 $0 \leqslant t \leqslant b, 0 \leqslant h \leqslant h_0$ 和所有的 $\mathbf{y}$, $\boldsymbol{\phi}(\mathbf{y}, t, h)$ 关于 $\mathbf{y}, t, h$ 连续,并且在该区域内对 $\mathbf{y}$ 满足 Lipschitz 条件,则收敛性的必要且充分条件如下:

$$\boldsymbol{\phi}(\mathbf{y}(t), t, 0) = \mathbf{f}(\mathbf{y}(t), t). \tag{4.10}$$

等式 (4.10) 称为相容条件. 由于通过适当的选取初值, 对给定的 $t, \mathbf{y}(t)$ 可以取任意值, (4.10) 将按形式

$$\boldsymbol{\phi}(\mathbf{y}, t, 0) = \mathbf{f}(\mathbf{y}, t)$$

对任意 y 成立.

证明: 令 $\boldsymbol{\phi}(\mathbf{y}, t, 0) = \mathbf{g}(\mathbf{y}, t)$.

因为 $\mathbf{g}$ 满足定理 4.1 的条件,微分方程

$$\mathbf{z}' = \mathbf{g}(\mathbf{z}, t), \quad \mathbf{z}_0 = y_0 \tag{4.11}$$

有唯一的可微解. 我们将证明由 (4.9) 给出的数值解收敛到 $\mathbf{z}(t)$, 因此 $\mathbf{f} = \mathbf{g}$ 是必要且充分的条件. 数值解满足

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h\boldsymbol{\phi}(\mathbf{y}_n, t_n, h). \tag{4.12}$$

利用中值定理

$$z^i(t_{n+1}) = z^i(t_n) + hg^i(\mathbf{z}(t_n + \theta^i h), t_n + \theta^i h),$$
$$0 < \theta^i < 1.$$

从 (4.2) 减去此式并令 $e_n = y_n - z(t_n)$, 我们得到

$$\begin{aligned} e_{n+1}^i = e_n^i + h[&\phi^i(\mathbf{y}_n, t_n, h) - \phi^i(\mathbf{z}(t_n), t_n, h) \\ &+ \phi^i(\mathbf{z}(t_n), t_n, h) - \phi^i(\mathbf{z}(t_n), t_n, 0) \\ &+ \phi^i(\mathbf{z}(t_n), t_n, 0) - g^i(\mathbf{z}(t_n + \theta^i h), t_n + \theta^i h)]. \end{aligned} \tag{4.13}$$

如定理 1.3,我们可以继续下去不需要附加的假设.但是,如果还假设 $\boldsymbol{\phi}$ 关于 t 和 h 满足 Lipschitz 条件, 如实际上将碰到的那样,则得到下面的估计:

$$\|\boldsymbol{\phi}(\mathbf{y}_n, t, h) - \boldsymbol{\phi}(\mathbf{z}(t_n), t, h)\| \leqslant L\|\mathbf{y}_n - \mathbf{z}(t_n)\| = L\|\mathbf{e}_n\|$$

$$\|\boldsymbol{\phi}(\mathbf{z}(t_n), t, h) - \boldsymbol{\phi}(\mathbf{z}(t_n), t_n, 0)\| \leqslant L_1 h$$

和

$$
\begin{aligned}
&|\{\psi^i(\mathbf{z}(t_n), t_n, 0) - g^i(\mathbf{z}(t_n + \theta^i h), t_n + \theta^i h)\}| \\
&= |\{g^i(\mathbf{z}(t_n), t_n) - g^i(\mathbf{z}(t_n + \theta^i h), t_n + \theta^i h)\}| \\
&\leqslant L|z'(t_n + \xi\theta^i h)|\theta^i h + L_3\theta^i h \leqslant L_4 h.
\end{aligned}
$$

因此，(4.13) 最后一行的模不超过 L_2h. 把这些代入 (4.13)，得到

$$
\begin{aligned}
\|e_{n+1}\| &\leqslant \|e_n\| + hL\|e_n\| + h^2(L_1 + L_2) \\
&= (1 + hL)\|e_n\| + h^2(L_1 + L_2). \qquad (4.14)
\end{aligned}
$$

这是引理 1.1 类型的差分方程. 我们有

$$
\|e_N\| \leqslant (L_1 + L_2)h\frac{e^{Lb-1}}{L} + e^{Lb}\|e_0\|.
$$

当 $h \to 0$ 和 $\|e_0\| \to 0$ 时，此式收敛于零，所以，数值解收敛到 (4.11) 的解，条件 $\mathbf{g}(\mathbf{y}, t) = \mathbf{f}(\mathbf{y}, t)$ 成立的充分性立即可得. 另一方面，如果收敛性成立，则(4.11)的解 $\mathbf{z}(t)$ 等同于 $\mathbf{y}'(t) = \mathbf{f}(\mathbf{y}(t), t)$ 的解 $y(t)$. 再假设 $\mathbf{f}$ 和 $\mathbf{g}$ 在某点 (y_a, t_a) 不同. 如果考虑从 (y_a, t_a) 开始计算的初值问题，有

$$
\begin{aligned}
\mathbf{y}'(t_a) = \mathbf{f}(\mathbf{y}(t_a), t_a) &\neq \mathbf{g}(\mathbf{y}(t_a), t_a) \\
&= \mathbf{g}(\mathbf{z}(t_a), t_a) = \mathbf{z}'(t_a).
\end{aligned}
$$

导致矛盾.

定理 4.3　应用的例子.

A. 四阶经典 Runge-Kutta 方法应用到一阶方程组 $\mathbf{y}' = \mathbf{f}(\mathbf{y})$.

给定 $\mathbf{f}$ 满足 Lipschitz 条件，则 $\mathbf{K}_0(\mathbf{y}) = h\mathbf{f}(\mathbf{y})$ 满足

$$
\|\mathbf{K}_0(\mathbf{y}) - \mathbf{K}_0(\mathbf{y}^*)\| \leqslant hL\|\mathbf{y} - \mathbf{y}^*\|,
$$

$$
\mathbf{K}_1(\mathbf{y}) = h\mathbf{f}\left(\mathbf{y} + \frac{1}{2}\mathbf{K}_0(\mathbf{y})\right) \text{满足}
$$

$$
\begin{aligned}
&\|\mathbf{K}_1(\mathbf{y}) - \mathbf{K}_1(\mathbf{y}^*)\| \\
&\leqslant hL\left\|\mathbf{y} - \mathbf{y}^* + \frac{1}{2}\mathbf{K}_0(\mathbf{y}) - \frac{1}{2}\mathbf{K}_0(\mathbf{y}^*)\right\|
\end{aligned}
$$

$$\leqslant hL\left(1+\frac{1}{2}hL\right)\|\mathbf{y}-\mathbf{y}^*\|,$$

$\mathbf{K}_2(\mathbf{y}) = h\mathbf{f}\left(\mathbf{y}+\frac{1}{2}\mathbf{K}_1(\mathbf{y})\right)$ 满足

$$\begin{aligned}
&\|\mathbf{K}_2(\mathbf{y})-\mathbf{K}_2(\mathbf{y}^*)\| \\
&\leqslant hL\left\|\mathbf{y}-\mathbf{y}^*+\frac{1}{2}\mathbf{K}_1(\mathbf{y})-\frac{1}{2}\mathbf{K}_1(\mathbf{y}^*)\right\| \\
&\leqslant hL\left(1+\frac{1}{2}hL\right)+\frac{1}{4}(hL)^2\|\mathbf{y}-\mathbf{y}^*\|,
\end{aligned}$$

和 $\mathbf{K}_3(\mathbf{y}) = h\mathbf{f}(\mathbf{y}+\mathbf{K}_2(\mathbf{y}))$ 满足

$$\begin{aligned}
\|\mathbf{K}_3(\mathbf{y})-\mathbf{K}_3(\mathbf{y}^*)\| &\leqslant hL\|\mathbf{y}-\mathbf{y}^*+\mathbf{K}_2(\mathbf{y})-\mathbf{K}_2(\mathbf{y}^*)\| \\
&\leqslant hL\left(1+hL+\frac{1}{2}(hL)^2+\frac{1}{4}(hL)^3\right)\|\mathbf{y}-\mathbf{y}^*\|.
\end{aligned}$$

所以，

$$\boldsymbol{\phi}(\mathbf{y},t,h)=\frac{1}{6h}(\mathbf{K}_0+2\mathbf{K}_1+2\mathbf{K}_2+\mathbf{K}_3)$$

满足

$$\begin{aligned}
&\|\boldsymbol{\phi}(\mathbf{y},t,h)-\boldsymbol{\phi}(\mathbf{y}^*,t,h)\| \\
&\leqslant \frac{L}{6}\Big(1+2+hL+2+hL+\frac{1}{2}(hL)^2 \\
&\quad +1+hL+\frac{1}{2}(hL)^2+\frac{1}{4}(hL)^3\Big)\|\mathbf{y}-\mathbf{y}^*\| \\
&= L\left(1+\frac{1}{2}hL+\frac{1}{6}(hL)^2+\frac{1}{24}(hL)^3\right)\|\mathbf{y}-\mathbf{y}^*\|.
\end{aligned}$$

因此，$\boldsymbol{\phi}$ 满足关于 $\mathbf{y}$ 的 Lipschitz 条件．还可看出，它对 h 是连续的，所以，可以得出方程组关于经典四阶 Runge-Kutta 方法收敛的结论．

B．二阶 Taylor 级数方法应用于二阶方程

$$y''=f(y,y',t),$$

其中 f, $\partial f/\partial y$, $\partial f/\partial t$ 满足关于 t, y 和 y' 的 Lipschitz 条件．

$$y_{n+1}=y_n+hy'_n+\frac{h^2}{2}f_n+\frac{h^3}{6}\frac{df_n}{dt},$$
$$y'_{n+1}=y'_n+hf_n+\frac{h^2}{2}\frac{df_n}{dt}. \quad (4.15)$$

如果向量 $\mathbf{y}$ 是 $[y, y']^T$, 那么

$$\boldsymbol{\phi}(\mathbf{y}, t, h)=\left[y'_n+\frac{h}{2}f_n+\frac{h^2}{2}\frac{df_n}{dt}, f_n+\frac{h}{2}\frac{df_n}{dt}\right]^T.$$

显然 $\boldsymbol{\phi}$ 满足关于 t, y 和 y' 的 Lipschitz 条件且 $\boldsymbol{\phi}(\mathbf{y}, t, 0)=[y'_n, f_n]^T$, 所以方法收敛(事实上,不需要 df/dt 的 Lipschitz 条件,只要有界性即可. 对于这种情况, 定理 4.3 的证明可以修改).

4.4. 误差界和收敛的阶

定理 4.3　给出了收敛的条件,但是没有详细说明解收敛的速度. 为了研究这点, 我们需要定义局部截断误差 $\mathbf{d}_n(h)$. 它由下式:

$$\mathbf{d}_n(h)=h\boldsymbol{\phi}(\mathbf{y}(t_n), t_n, h)-(\mathbf{y}(t_{n+1})-\mathbf{y}(t_n)) \quad (4.16)$$

给出.因此,它是使微分方程的解未能满足数值方法中应用的方程的一个量. 我们看到, $\mathbf{d}_n(h)$ 能够表示成 h 的幂级数,其系数是解的导数的多项式. 利用推导 $\mathbf{d}_n(h)$ 的 Taylor 级数展开的余项, 界就可以得到. 如果对所有的 $0\leqslant h\leqslant h_0$ 和所有的 t 以及所考虑的 $\mathbf{y}$, 有如下形式的界:

$$\|\mathbf{d}_n(h)\|\leqslant Dh^{r+1}, \quad (4.17)$$

则可证明全体误差是 r 阶的. 这样, 就说方法具有阶 r.

定理 4.4.　如果 $\boldsymbol{\phi}$ 满足定理 4.3 的条件, 而且由 (4.16) 定义的 $\mathbf{d}_n(h)$ 满足 (4.17), 那么误差有如下的界:

$$\|\mathbf{y}_n-\mathbf{y}(t_n)\|\leqslant Dh^r\frac{e^{Lb}-1}{L}+e^{Lb}\|\mathbf{y}_0-\mathbf{y}(t_0)\|. \quad (4.18)$$

证明: 如果记 $\mathbf{e}_n=\mathbf{y}_n-\mathbf{y}(t_n)$ 且从 (4.12) 的两边减去

$\mathbf{y}(t_{n+1})$，我们得到

$$\mathbf{e}_{n+1} = \mathbf{y}_n - \mathbf{y}(t_n) + [h\boldsymbol{\phi}(\mathbf{y}_n, t_n, h) - (\mathbf{y}(t_{n+1}) - \mathbf{y}(t_n))] = \mathbf{e}_n + h[\boldsymbol{\phi}(\mathbf{y}_n, t_n, h) - \boldsymbol{\phi}(\mathbf{y}(t_n), t_n, h)] + \mathbf{d}_n(h). \tag{4.19}$$

所以，

$$\|\mathbf{e}_{n+1}\| \leqslant \|e_n\| + hL\|\mathbf{e}_n\| + h^{r+1}D = (1 + hL)\|\mathbf{e}_n\| + h^{r+1}D.$$

从引理 1.1 直接得到 (4.18).

我们看到，全体误差比局部截断误差低一阶.

应用定理 4.4 的例子.

考虑等式 (4.15) 确定的方法. 如果 $y(t)$ 有一连续的四阶导数，我们可以表示 $\mathbf{d}_n(h)$ 为

$$\left[-\frac{h^4}{24}y^{(4)}(\xi_n), -\frac{h^3}{6}y^{(4)}(\xi_n')\right]^T.$$

因此，对于 $h \leqslant h_0$，$\|\mathbf{d}_n(h)\| \leqslant h^3D$，其中 D 与 $\max|y^{(4)}|$ 成正比. 即使 y_n 的局部截断误差为 $O(h^4)$，收敛的总体速度也是二次的. 在例子

$$y'' = y, \quad y(0) = 1, \quad y'(0) = 1$$

表 4.1. $\mathbf{y}'' = \mathbf{y}$ 的积分

h	$y_h(1)$	$e = y_h(1) - 2.71828$	$\frac{e}{h^2}$	$y_h'(1)$	$e' = y_h'(1) - 2.71828$	$\frac{e'}{h^2}$
1	2.6666667	−0.0516152	−0.0516	2.5000000	−0.2182818	−0.2183
$\frac{1}{2}$	2.6979167	−0.0203652	−0.0815	2.6510417	−0.0672402	−0.2690
$\frac{1}{4}$	2.7118655	−0.0064163	−0.1027	2.6997631	−0.0185187	−0.2963
$\frac{1}{8}$	2.7164846	−0.0017973	−0.1150	2.7134331	−0.0048487	−0.3103
$\frac{1}{16}$	2.7178066	−0.0004752	−0.1217	2.7170421	−0.0012398	−0.3174
$\frac{1}{32}$	2.7181597	−0.0001222	−0.1251	2.7179684	−0.0003134	−0.3209
$\frac{1}{64}$	2.7182509	−0.0000310	−0.1269	2.7182031	−0.0000788	−0.3227

中，我们可以看到这点，如表 4.1 所示，在 $t = O(h)1$ 上积分，$h = 2^{-m}(m = 0, 1, 2, \cdots, 6)$. y 和 y' 两者的误差按 $O(h^2)$

变化.

定理 4.4 还考虑到给舍入误差积累的一误差界. 它是这样处理的: 在 $h\phi$ 的定义中,包含舍入误差, 使得 $\mathbf{d}_n(h)$ 的定义 (4.16) 包含舍入和截断误差.只要能够给出例如 (4.17) 的界, 定理 4.4 即可应用.

4.5. 渐近误差的估计

在 Euler 方法的情形,误差界不是误差大小的精确估计,一般情形也是如此. 在上一节最后的例中,我们有

$$\mathbf{d}_n(h) = \left[-\frac{h^4}{24} y^{(4)}(\xi_n), -\frac{h^3}{6} y^{(4)}(\xi_n)\right]^T.$$

利用最大模,对于 $0 \leqslant t \leqslant 1$, $0 \leqslant h \leqslant 1$, 我们得到

$$\|\mathbf{d}_n(h)\| \leqslant h^3 \max_{0 \leqslant t \leqslant 1} \frac{y^{(4)}(t)}{6}.$$

因为 $y(t) = e^t$, $D = e/6$,

$$\frac{\partial \phi}{\partial(y, y')} = \begin{bmatrix} \frac{h}{2} & 1 + \frac{h^2}{6} \\ 1 & \frac{h}{2} \end{bmatrix},$$

所以, 对于 $h \leqslant 1$ 有 $L = 1 + (h/2) + (h^2/6) \leqslant \frac{5}{3}$.

利用这些数字, 由 (4.18) 给出的误差界是

$$h^2 \frac{e}{10}(e^{5/3} - 1) \cong 1.167h^2.$$

从表 4.1 看到, y 和 y' 上的误差大约渐近到 $0.128h^2$ 和 $0.325h^2$, 因此,界大了三倍多. 在复杂的问题中,界可能更大.

鉴于此,我们再次考察当 $h \to 0$ 时渐近形式的误差估计.如果我们能够把局部截断误差写成

$$\begin{aligned} h\phi(\mathbf{y}(t), t, h) - (\mathbf{y}(t+h) - \mathbf{y}(t)) &= \mathbf{d}_n(h) \\ &= h^{r+1}\boldsymbol{\phi}(\mathbf{y}, t) + O(h^{r+2}) \end{aligned} \tag{4.20}$$

[如果 $\boldsymbol{\phi}$ 和 $\mathbf{f}$ 有连续的 $(r+1)$ 阶导数,即可这样做],那么,$\boldsymbol{\phi}$ 被称为主误差函数.从定理 4.4 知,误差是 $O(h^r)$.于是我们寻求如下形式的误差表示式:

$$\mathbf{e}_n = h^r\boldsymbol{\delta}(t_n) + O(h^{r+1}).$$

为了研究 $\boldsymbol{\delta}(t)$ 项,我们开始把

$$\mathbf{e}_n = h^r\boldsymbol{\delta}_n$$

代入 (4.19),得到

$$\begin{aligned}\boldsymbol{\delta}_{n+1} = \boldsymbol{\delta}_n + h^{1-r}[\boldsymbol{\phi}(\mathbf{y}(t_n) + h^r\boldsymbol{\delta}_n, t_n, h) - \boldsymbol{\phi}(\mathbf{y}(t_n), t_n, h)] \\ + h\phi(\mathbf{y}(t_n), t_n) + O(h^2).\end{aligned} \qquad (4.21)$$

如果假设 $\boldsymbol{\phi}$ 关于它的变量 $\mathbf{y}$ 和 h 在规定区域内二次连续可微并且导数有界(由二阶导数的连续性推导出这点),我们能够写出:

$$\begin{aligned}&\boldsymbol{\phi}(\mathbf{y}(t_n) + h^r\boldsymbol{\delta}_n, t_n, h) \\ &\quad = \boldsymbol{\phi}(\mathbf{y}(t_n), t_n, h) + \boldsymbol{\phi}_y(\mathbf{y}(t_n), t_n, h)h^r\boldsymbol{\delta}_n \\ &\qquad + \frac{1}{2}\boldsymbol{\phi}_{yy}(\mathbf{y}(t_n) + \xi, t_n, h)h^{2r}\boldsymbol{\delta}_n\boldsymbol{\delta}_n,\end{aligned}$$

其中 $\boldsymbol{\phi}_y$ 是矩阵,其分量是 $\partial\phi^i/\partial y^j$,$\boldsymbol{\phi}_{yy}$ 是一三阶张量[1],其分量是 $\partial^2\phi^i/\partial y^j\partial y^k$, 且 $\boldsymbol{\xi}$ 是一向量,其每一个分量小于 $\mathbf{e}_n$ 对应的分量. 因为已经假设二阶导数有界, 且已证明(定理 4.3) $\|\boldsymbol{\delta}_n\|$ 有界, 所以,最后一项可以写成 $\mathbf{K}_1h^{2r}$, 其中 $\|\mathbf{K}_1\|$ 是有界的. 利用中值定理,可以将第二项表示成

$$\begin{aligned}&\boldsymbol{\phi}_y(\mathbf{y}(t_n), t_n, h)h^r\boldsymbol{\delta}_n \\ &\quad = \boldsymbol{\phi}_y(\mathbf{y}(t_n), t_n, 0)h^r\boldsymbol{\delta}_n + \boldsymbol{\phi}_{yh}(\mathbf{y}(t_n), t_n, \xi')h^{r+1}\boldsymbol{\delta}_n.\end{aligned}$$

再利用二阶导数的有界性, 最后一项形如 $\mathbf{K}_2h^{r+1}$, 其中 $\|\mathbf{K}_2\|$ 是有界的. 最后, 注意 $\boldsymbol{\phi}(\mathbf{y}, t, 0) = \mathbf{f}(\mathbf{y}, t)$, 所以 $\boldsymbol{\phi}_y(\mathbf{y}, t,$

---

1) 简单说来, 张量是广义矩阵. 我们仅注意 $\boldsymbol{\phi}_{yy}\boldsymbol{\delta}\boldsymbol{\delta}$ 是一向量, 其分量是 $\sum_{jk}(\partial^2\phi^i/\partial y^j\partial y^k)\delta_j\delta_k$, 或用前面的记法 $\phi^i_{jk}\delta^j\delta^k$.

$0)=\mathbf{f}_y(\mathbf{y},t)$. 代入 (4.21)，我们得到

$$\boldsymbol{\delta}_{n+1}=\boldsymbol{\delta}_n+h[\mathbf{f}_y(\mathbf{y}(t_n),t_n)\boldsymbol{\delta}_n+\boldsymbol{\phi}(\mathbf{y}(t_n),t_n),\\ +h\mathbf{K}_2+h^r\mathbf{K}_1]. \tag{4.22}$$

如果把 (4.22) 看成解微分方程

$$\boldsymbol{\delta}'(t)=\mathbf{f}_y(\mathbf{y}(t),t)\boldsymbol{\delta}(t)+\boldsymbol{\phi}(\mathbf{y}(t),t),\ \boldsymbol{\delta}(0)=\frac{\mathbf{e}_0}{h^r} \tag{4.23}$$

的一个数值方法，则方法的增量满足定理 4.3 的条件，因此，$\boldsymbol{\delta}_n$ 收敛到 $\boldsymbol{\delta}(t_n)$ 且满足定理 4.4 的条件及 $r=1$ (在那定理中)，并有

$$D=\max_i\frac{1}{2}\|\boldsymbol{\delta}''(t)\|+h_0^{r-1}\max\|\mathbf{K}_1\|+\max\|\mathbf{K}_2\|,$$

对于 $0\leqslant h\leqslant h_0$. 因此，我们证明了下面的定理：

定理 4.5. 如果截断误差能够表示成 (4.20) 且 $\boldsymbol{\phi}$ 有连续的二阶导数，则误差满足

$$\mathbf{e}_n=h^r\boldsymbol{\delta}(t_n)+O(h^{r+1}), \tag{4.24}$$

其中 $\boldsymbol{\delta}(t)$ 是 (4.23) 的解.

这个定理很少用来作为直接估计误差的办法，因为计算 $\mathbf{f}_y$ 和 $\boldsymbol{\phi}$ 对大多数方程和有兴趣的方法来说是太复杂了. 然而，它常用于证明间接的误差估计，如外推到极限那样. 如同应用 Euler 方法于单个方程那样，我们可以用两个不同的步长 h 和 qh 计算得到 $\mathbf{y}(b)$ 的估计. 称这些值为 $\mathbf{y}_h(b)$ 和 $\mathbf{y}_{qh}(b)$. 由 (4.24) 有

$$\mathbf{y}_h(b)=\mathbf{y}(b)+h^r\boldsymbol{\delta}(b)+O(h^{r+1}),$$
$$\mathbf{y}_{qh}(b)=\mathbf{y}(b)+q^rh^r\boldsymbol{\delta}(b)+O(h^{r+1})$$

或

$$\boldsymbol{\delta}(b)=h^{-r}\frac{y_h(b)-y_{qh}(b)}{1-q^r}+O(h).$$

另外，我们用

$$y(b)=\frac{y_{qh}(b)-q^ry_h(b)}{1-q^r}+O(h^{r+1})$$

能得到 $y(b)$ 的较好的近似值. 对于 $q=\frac{1}{2}$ 应用表 4.1，其结果在表 4.2 中给出. 现在有 $O(h^3)$ 的收敛性.

应用定理 4.5 的例子.

我们再一次考虑用于产生表 4.1 的二阶 Taylor 级数方法. 我们有

$$\mathbf{d}_n(h)=h^3\left[0,\ -\frac{e^{t_n}}{6}\right]^T+O(h^4),$$

于是

$$\boldsymbol{\phi}(\mathbf{y},t)=\left[0,\ -\frac{e^t}{6}\right]^T.$$

微分方程是 $y''=y$，我们把它写作

$$y'=\begin{bmatrix}y\\y'\end{bmatrix}'=\begin{bmatrix}y'\\y''\end{bmatrix}=\mathbf{f}(\mathbf{y},t),$$

那么

$$\mathbf{f}_y=\begin{bmatrix}0&1\\1&0\end{bmatrix}.$$

于是 $\boldsymbol{\delta}(t)$ 满足

$$\begin{bmatrix}\delta^1(t)\\\delta^2(t)\end{bmatrix}'=\begin{bmatrix}0&1\\1&0\end{bmatrix}\begin{bmatrix}\delta^1(t)\\\delta^2(t)\end{bmatrix}-\begin{bmatrix}0\\\dfrac{e^t}{6}\end{bmatrix}.$$

它有解

$$\delta^1(t)=-\frac{1}{24}(2te^t+e^{-t}-e^t),$$

$$\delta^2(t)=-\frac{1}{24}(2te^t+e^t-e^{-t}).$$

因此,

$$\delta^1(1)=-\frac{1}{24}(e+e^{-1})\cong-0.1286,$$

$$\delta^2(1) = -\frac{1}{24}(3e - e^{-1}) \cong -0.3245.$$

它与表 4.1 中的误差相当一致.

表 4.2. 用二阶 Taylor 方法，方程 $y'' = y$ 的结果的外推

h	$y_h(1)$	$\dfrac{4y_{h/2}(1) - y_h(1)}{3}$	误差	误差$/h^3$
1	2.6666667	2.7083333	−0.00994850	−0.0796
$\frac{1}{2}$	2.6979167	2.7165152	−0.00176668	−0.1131
$\frac{1}{4}$	2.7118655	2.7180242	−0.00025760	−0.1319
$\frac{1}{8}$	2.7164846	2.7182473	−0.00003455	−0.1415
$\frac{1}{16}$	2.7178066	2.7182774	−0.00000446	−0.1463
$\frac{1}{32}$	2.7181597	2.7182813	−0.00000057	−0.1487
$\frac{1}{64}$	2.7182509	2.7182818	−0.00000007	−0.1498
$\frac{1}{128}$	2.7182740	—	—	—
h	$y'_h(1)$	$\dfrac{4y'_{h/2}(1) - y'_h(1)}{3}$	误差	误差$/h^3$
1	2.5000000	2.7013889	−0.01689294	−0.1351
$\frac{1}{2}$	2.6510417	2.7160036	−0.00227827	−0.1458
$\frac{1}{4}$	2.6997631	2.7179898	−0.00029206	−0.1495
$\frac{1}{8}$	2.7134331	2.7182451	−0.00003678	−0.1506
$\frac{1}{16}$	2.7170421	2.7182772	−0.00000461	−0.1509
$\frac{1}{32}$	2.7179684	2.7182813	−0.00000058	−0.1510
$\frac{1}{64}$	2.7182031	2.7182818	−0.00000007	−0.1510
$\frac{1}{128}$	2.7182621	—	—	—

4.5.1. 由数值近似产生的扰动

数值解 $\mathbf{y}_n$ 可以表示成 $\mathbf{z}(t_n)$，这里 $\mathbf{z}$ 是扰动微分方程的解. 做与 1.3.5 中类似的分析. 定义解 $\mathbf{y}(t_n+\tau, \mathbf{y}_n, t_n)$ 的局部误差为

$$\begin{aligned}\mathbf{d}(\tau;\mathbf{y}_n, t_n) &= \tau^{r+1}\mathbf{T}(\tau;\mathbf{y}_n, t_n)\\ &= \mathbf{y}_n + \tau\boldsymbol{\phi}(\mathbf{y}'_n, t_n, \tau) - \mathbf{y}(t_n+\tau; \mathbf{y}_n, t_n),\end{aligned}$$

且令 $\mathbf{z}(t)$ 在 $[t_n, t_{n+1}]$ 上由

$$\mathbf{z}(t_n+\tau)=\frac{\tau R}{h}+h^r\tau\mathbf{T}(\tau;\mathbf{y}_n,t_n)+\mathbf{y}(t_n+\tau;\mathbf{y}_n,t_n)$$

所定义，然后残差由

$$\begin{aligned}\mathbf{r}(t_n+\tau)&=\mathbf{z}'(t_n+\tau)-\mathbf{f}(\mathbf{z}(t_n+\tau),t_n+\tau)\\&=\frac{R}{h}+h^rT(\tau;\mathbf{y}_n,t_n)+h^r\tau\frac{d}{d\tau}T(\tau;\mathbf{y}_n,t_n)\\&\quad+\mathbf{f}(\mathbf{y}(t_n+\tau;y_n,t_n),t_n+\tau)\\&\quad-\mathbf{f}(\mathbf{z}(t_n+\tau),t_n+\tau)\end{aligned}\tag{4.25}$$

给出．对任何给定的方法，我们能够估计 $\|\mathbf{r}(t)\|$ 的界或者得到很好的近似，因此，对一阶方程组的 r 阶 Taylor 级数方法，

$$\begin{aligned}\mathbf{d}(\tau;\mathbf{y}_n,t_n)=&-\frac{\tau^{r+1}}{(r+1)!}\frac{d^{r+1}}{dt^{r+1}}\mathbf{y}(t_n;\mathbf{y}_n,t_n)\\&-\frac{\tau^{r+2}}{(r+2)!}\frac{d^{r+2}}{dt^{r+2}}\mathbf{y}(\xi_\tau;\mathbf{y}_n,t_n).\end{aligned}$$

同时

$$\begin{aligned}\frac{d}{d\tau}\mathbf{d}(\tau;y_n,t_n)&=(r+1)\tau^rT(\tau;y_n,t_n)\\&+\tau^{r+1}\frac{d}{d\tau}T(\tau;\mathbf{y}_n,t_n)=-\frac{\tau^r}{r!}\frac{d^{r+1}}{dt^{r+1}}\mathbf{y}(t_n;\mathbf{y}_n,t_n)\\&-\frac{\tau^{r+1}}{(r+1)!}\frac{d^{r+2}}{dt^{r+2}}\mathbf{y}(\xi_\tau;\mathbf{y}_n,t_n).\end{aligned}$$

如果 $(r+2)$ 阶导数不超过M，$\mathbf{T}(\tau;y_n,t_n)$不超过T，又 $\mathbf{f}$ 的 Lipschitz 常数是 L，我们得到

$$\begin{aligned}&\left\|\mathbf{r}(t_n+\tau)-\frac{R}{h}+\frac{h^ry_n^{(r+1)}}{(r+1)!}\right\|\\&\leqslant\left(LT+\frac{2r+2}{(r+2)!}M\right)h^{r+1}.\end{aligned}\tag{4.26}$$

$y_n^{(r+1)}$ 对计算出的值 y_n 的微分方程的解来求值，实际上 $\mathbf{y}_n$ 是我们能估计的唯一的值，如果把一般的 r 阶单步方法的局部

截断误差写成

$$h^{r+1}\boldsymbol{\phi}(\mathbf{y}, t) + h^{r+2}\tilde{T}(h, \mathbf{y}, t),$$

且假设 $\tilde{T}$ 对 h 有连续的有界导数，我们可以类似证明残差满足

$$\left\|\mathbf{r}(t_n+\tau) - \frac{R}{h} - h^r\boldsymbol{\phi}(\mathbf{y}_n, t_n)\right\| = O(h^{r+1}).$$

此式可用于给出误差界或估计误差，于是可以得到定理 4.5 的结果，其中 ϕ 沿计算的解在 $\mathbf{z}(t)$ 和 t 求值.

4.6. 误差界和估计定理的一般应用

定理 4.4 和 4.5 是用增量函数 $\boldsymbol{\phi}$ 上的条件来叙述的，这些条件包括导数的存在性、连续性和有界性，以及局部截断误差的渐近形式．实际上，已知关于微分方程 $\mathbf{f}$ 的一些情况，并且必须把它同 $\boldsymbol{\phi}$ 联系起来．这一节研究这种关系，目的是了解给定的方法所要求的性质，即要求 $\mathbf{f}$ 有哪些足够的条件.

首先注意，虽然叙述存在性定理 4.1 是利用 $\mathbf{y}$ 子空间的无界区域，但我们常常可以在 $0 \leqslant t \leqslant b$ 和 $0 \leqslant h \leqslant h_0$ 的有界闭区域上进行讨论．因此，导数的连续性对于其有界性是充分的．我们要求 $\mathbf{f}$ 的连续性条件，它也是有界的．此外，假设 $\mathbf{f}$ 的一阶偏导数存在且连续，从微分方程有

$$\mathbf{y}'' = \frac{d}{dt}(\mathbf{f}(\mathbf{y}, t)) = \frac{\partial \mathbf{f}}{\partial t} + \frac{\partial \mathbf{f}}{\partial \mathbf{y}}\mathbf{y}'.$$

由于存在性定理，$\mathbf{y}'$ 存在且连续，因此，$\mathbf{y}''$ 存在且连续．显然，我们可以重复这个过程并且叙述如下：

如果 $\mathbf{f}(\mathbf{y}, t)$ 关于它的所有变量 q 次连续可微，那么 $\mathbf{y}^{(q+1)}$ 存在且连续.

在许多应用中，$\mathbf{f}(\mathbf{y}, t)$ 在所考虑的区域内是解析的．但是，有些重要的应用中导数也可以间断．在此种情形，解在其

导数上有间断，这就使定理 4.4 和 4.5 成立而需要的导数连续的条件变得没有意义．如果连续导数的最高阶是 q，方法的阶是 $r \geqslant q-1$，那么定理 4.4 证明了 $q-1$ 阶方法的收敛性．然而定理 4.5 是不可应用的．

截断误差由 (4.16) 定义．如果希望得到形如 (4.17) 的界，可以利用含有 h^{r+1} 的余项的 Taylor 级数将 $\mathbf{d}_n(h)$ 展成 h 的幂级数．为此，我们要求 $\boldsymbol{\phi}$ 有连续的 r 阶 h 次导数，且 $\mathbf{y}$ 是 $(r+1)$ 次连续可微的．如果希望导出渐近形式 (4.20)，可利用带有余项且终止在 h^{r+2} 的 Taylor 级数来展开．为此，需要 $\boldsymbol{\phi}$ 对 h 的连续 $(r+1)$ 次导数和 $\mathbf{y}$ 的连续 $(r+2)$ 次导数，因此，可以写

$$\begin{aligned}\boldsymbol{\phi}(\mathbf{y}, t) &= \frac{1}{r!}\frac{\partial^r}{\partial h^r}[h^r\boldsymbol{\phi}(\mathbf{y}, t) + O(h^{r+1})]_{h=0}\\ &= \frac{1}{r!}\frac{\partial^r}{\partial h^r}\left[\frac{d_n(h)}{h}\right]_{h=0} = \frac{1}{r!}\frac{\partial^r\boldsymbol{\phi}}{\partial h^r}(y, t, h)\Big|_{h=0}\\ &\quad - \frac{1}{(r+1)!}\frac{d^r}{\partial t^r}\mathbf{f}(\mathbf{y}, t).\end{aligned} \tag{4.27}$$

利用这些说明，可以考察前几章讨论的两种类型的单步方法．

4.6.1. Taylor 级数方法

在此种情形，用

$$\boldsymbol{\phi}(\mathbf{y}, t, h) = \sum_{q=0}^{r-1}\frac{h^q}{(q+1)!}\frac{d^q}{dt^q}\mathbf{f}(\mathbf{y}, t)$$

定义 r 阶方法．如果 $\mathbf{f}$ 有连续的 r 阶导数，显然 $\boldsymbol{\phi}$ 满足定理 4.4 的条件，其中

$$D = \frac{1}{(r+1)!}\max_{t,\mathbf{y}}\left\|\frac{d^r}{dt^r}\mathbf{f}(\mathbf{y}, t)\right\|.$$

如果 $\mathbf{f}$ 是 $r+1$ 次连续可微的，那么，我们可以写

$$\phi(\mathbf{y}, t) = -\frac{1}{(r+1)!}\mathbf{y}^{(r+1)}.$$

因 $r \geqslant 1$, $\boldsymbol{\phi}$ 对它的所有变量二次连续可微，所以，定理 4.5 成立.

4.6.2. Runge-Kutta 方法

显式 Runge-Kutta 方法由逐次代入函数 $\mathbf{f}$ 组成，因此，$\boldsymbol{\phi}$ 同 $\mathbf{f}$ 有相同的可微次数. 如果 $\mathbf{f}$ 是 r 次连续可微的，则存在 D，使定理 4.4 成立. 如果 $\mathbf{f}$ 是 $r+1$ 次连续可微的，那么，定理 4.5 成立. 但是，$\boldsymbol{\phi}$ 的求值一般不是简单的问题.

在中点法则的情形，可以利用等式 (4.27) 得到

$$\boldsymbol{\phi} = \frac{1}{2}\frac{\partial^2}{\partial h^2}\boldsymbol{\psi}(\mathbf{y}, t, h)_{h=0} - \frac{1}{6}\frac{d^2}{dt^2}\mathbf{f},$$

其中

$$\psi^i = f^i\left(\mathbf{y} + \frac{h}{2}\mathbf{f}(\mathbf{y})\right).$$

于是有

$$\frac{\partial \psi^i}{\partial h} = \frac{1}{2} f^i_j\left(\mathbf{y}, \frac{h}{2}\mathbf{f}(\mathbf{y})\right) f^j$$

和

$$\frac{1}{2}\frac{\partial^2 \psi^i}{\partial h^2}\bigg|_{h=0} = \frac{1}{8} f^i_{jk} f^j f^k.$$

由此得

$$\phi^i = -\frac{1}{24} f^i_{jk} f^j f^k - \frac{1}{6} f^i_j f^j_k f^k.$$

与等式 (2.11) 相同.

隐式 Runge-Kutta 方法对足够小的 h 有同样的性质. $\boldsymbol{\phi}$ 依赖于非线性方程 (2.23) 的解. 2.5.2 指出，当 $h \leqslant h_0$ 和 h_0 足够小时，由方程 (2.26) 给出的逐次代入方法收敛. 在这区

域内，像在等式(2.24)中的代入法一样，可以把解写成 h 的幂级数. 在 ϕ 中 h^r 的系数包含直到 $\mathbf{f}$ 的 r 阶导数，因此，ϕ 是关于 h 连续可微直到像 $\mathbf{f}$ 关于 $\mathbf{y}$ 和 t 可微的同样的阶. 定义 ϕ 关于 $\mathbf{y}$ 和 t 的关系的微分表明，如果 $\mathbf{f}$ 有连续的二阶导数，则对于足够小的 h，ϕ 关于它的所有变量有连续的二阶导数.

4.6.3. 对连续导数的要求

为使定理 4.4 和 4.5 成立，是否需要 r 阶或 $r+1$ 阶的连续导数. 这种问题是否合理? 需视情况而定. 许多问题在孤立点上某些导数有简单间断，如果这些点永远不在同一积分步长内，那么，间断可以忽略(但是，如果这些导数需要求值，必须注意，在间断点左边和右边的区间上积分时，用适当的值). 另一方面，某些间断不能忽略. 考虑 $y' = 2.5t^{3/2}$, $t \in [0, 1]$, $y(0) = 0$. 表 4.3 列出了应用梯形公式和经典 Runge-Kutta 方法的结果. 它们分别是二阶和四阶的方法. 我们看到，Runge-Kutta 的误差不是 $O(h^4)$，因为解的三阶导数在 $t=0$ 不存在.

表 4.3. 对于 t 从 0 到 1, $y' = 2.5t^{3/2}$ 的积分

h	梯形公式	误差 $/h^2$	四阶 Runge-Kutta	误差 $/h^4$	误差 $/h^{5/2}$
1	1.250000	0.2500	1.00592232	0.00592	0.005922
$\frac{1}{2}$	1.066942	0.2678	1.00107979	0.01728	0.006108
$\frac{1}{4}$	1.017545	0.2807	1.00019312	0.04944	0.006180
$\frac{1}{8}$	1.004531	0.2900	1.00003428	0.14043	0.006206
$\frac{1}{16}$	1.001159	0.2966	1.00000607	0.39778	0.006215
$\frac{1}{32}$	1.000294	0.3012	1.00000107	1.12568	0.006219
$\frac{1}{64}$	1.000074	0.3045	1.00000019	3.18462	0.006220

4.7. 变步长

定理 4.3 到 4.5 是建立在区间被分割成等长度区间的基

础之上的．实际上，考虑到解的变化性态，在区间上要调整步长．选取步长的办法在下一章讨论．我们要说明这一章的结果对变步长也适用，以此结束单步方法基础理论的讨论．

定理讨论了收敛性作为 $h\to 0$ 的一个函数．现在规定 h 作为最大步长且假设存在一函数 $\theta(t)$，对于 $t\in[0, b]$ 有 $0<\Delta\leqslant\theta(t)\leqslant 1$ 且使从 t_n 到 t_{n+1} 的步长是

$$h_n = h\theta(t_n),$$
$$t_{n+1} = t_n + h_n. \tag{4.28}$$

如果 $h>0$，由于 $h_n\geqslant\Delta h>0$，有限步将覆盖区间 $[0, b]$．然后可以证明，定理 4.3 和 4.4 仍然正确，而定理 4.5 用

$$\boldsymbol{\delta}'(t) = \frac{\partial \mathbf{f}}{\partial \mathbf{y}}\boldsymbol{\delta}(t) + \theta^r(t)\boldsymbol{\phi}(\mathbf{y}(t), t) \tag{4.29}$$

定义推广的渐近误差进行修改．

定理 4.3 的证明：

在以前的证明中，h 必须用 $\theta(t_n)h$ 代替．这样，方程 (4.14) 改写成

$$\|\mathbf{e}_{n+1}\| \leqslant (1+\theta(t_n)hL)\|\mathbf{e}_n\| + \theta^2(t_n)h^2(L_1+L_2). \tag{4.30}$$

因为 $|\theta(t_n)|\leqslant 1$，最后一项可以不超过 $kh^2\theta(t_n)$．我们现在证明，如果 $\|\mathbf{e}_0\|=0$，那么，由 (4.30) 推出

$$\|\mathbf{e}_n\| \leqslant \frac{\prod_{i=0}^{n-1}[1+hL\theta(t_i)](L\|e_0\|+kh)-kh}{L}. \tag{4.31}$$

对于 $n=0$，这是真确的．因此可用归纳法考察 $\|\mathbf{e}_{n+1}\|$．从 (4.30) 得到

$$\|\mathbf{e}_{n+1}\| \leqslant (1+\theta(t_n)hL)$$
$$\times\frac{\prod_{0}^{n-1}[1+\theta(t_n)hL](L\|\mathbf{e}_0\|+kh)-kh}{L} + kh^2(t_n)$$

$$= \frac{\prod_{0}^{n} [1+\theta(t_n)hL](L\|\mathbf{e}_0\| + kh) - kh}{L}.$$

这证明了 (4.31) 对所有 n 真确. 最后, 我们注意 $1 + \theta(t_n)hL \leqslant e^{\theta(t_n)hL}$, 有

$$\prod_{0}^{n-1} (1+\theta(t_n)hL) \leqslant e^{\sum_{0}^{n-1}\theta(t_n)L} = e^{t_nL} \leqslant e^{bL}.$$

因此, 象在前面的证明一样, 数值解收敛到 $\mathbf{z}(t)$ 且结论随之而得.

定理 4.4 的证明:

我们再从

$$\|\mathbf{e}_{n+1}\| \leqslant \|\mathbf{e}_n\|(1+\theta(t_n)hL) + \theta^{r+1}(t_n)h^{r+1}D$$

继续证明. 通过直接模拟 (4.31) 得到

$$\|\mathbf{e}_n\| \leqslant \frac{\prod_{0}^{n-1} [1 + hL\theta(t_i)](L\|\mathbf{e}_0\| + Dh^r) - Dh^r}{L}$$

$$\leqslant Dh^r\frac{e^{bL}-1}{L} + e^{bL}\|\mathbf{e}_0\|.$$

定理 4.5 的证明:

方程 (4.22) 之前仿照前面的证明, 而 (4.22) 则为

$$\boldsymbol{\delta}_{n+1} = \boldsymbol{\delta}_n + \theta(t_n)h[\mathbf{f}_y(\mathbf{y}(t_n), t_n)\boldsymbol{\delta}_n + \theta^r(t_n)\boldsymbol{\phi}(\mathbf{y}(t_n), t) + \theta(t_n)h\mathbf{K}_2 + \theta^r(t_n)h^r\mathbf{K}_1].$$

这可以看成是对于满足定理 4.3 和 4.4 的要求解微分方程 (4.29) 的变步长 Euler 方法(对于变步长, 已经证明), 结果立即可得.

问　　题

1. 证明: 如果 $\mathbf{f}(\mathbf{y})$ 对它的每个变量分别满足 Lipschitz 条件, 那么,

对任何一种模，有

$$\|\mathbf{f}(\mathbf{y}) - \mathbf{f}(\mathbf{y}^*)\| \leqslant L\|\mathbf{y} - \mathbf{y}^*\|.$$

反之亦然。

2. 证明：如果 f 关于 y, y' 和 t 满足 Lipschitz 条件，又 $\frac{df}{dt}$ 关于 y, y' 和 t 仅仅是连续的，则由等式(4.15)所定义的方法收敛。

3. 采用 Heun 方法，应用定理 4.3，4.4 和 4.5 于微分方程

$$y' = -2y + z + e^t,$$
$$z = 3y,$$
$$y(0) = 1.5, \quad z(0) = -0.25.$$

把你的结果同所计算的在 $t = 1$ 的结果进行比较，对于 $h = 2^{-m}$ $(m = 0, 1, \cdots, 6)$。

4. 如果 y 的误差小于 10^{-8}，使用二阶 Taylor 级数方法，在区间 $t \in [0,1]$ 上积分

$$y'' - y' + 2y = e^t,$$
$$y(0) = y'(0) = 2,$$

该用多大的固定步长？对于这个步长，y' 的误差近似于多少？

5. 证明 4.6 第二段的命题在有界闭区域上的正确性（可以假设存在性定理真确且 Lipschitz 常数存在，证明存在一个包含微分方程的解且对于 $0 \leqslant h \leqslant h_0$ 包含计算的所有近似值的有限区域是必要的）。

6. 证明：如果 $\mathbf{f}$ 二阶连续可微，又 h 充分小，隐式 Runge-Kutta 方法的增量函数 ϕ 关于 y, h 和 t 有连续的二阶导数。

7. 如果步长的最好选取是使得局部截断误差的大小 是步长的常数倍，则对于微分方程

$$y' = (e^{-t} + \sin t - y) + \cos t, \qquad y(0) = 1,$$

利用中点法则，应该使用什么样的函数 $\theta(t)$ 来给出变步长？在这些条件下，误差的渐近形式如何？

8. 说明表 4.3 最后一列的性质。

5. 步长和阶的选取

我们已经讨论了各种不同类型和阶的单步方法．对一个特殊问题，在选取求解的方法时，必须在不同类型的方法中选取，然后选取方法的阶和步长．这些参数的任何一个在整个积分中保持常数或者按照问题的性质而变化．假如一参数应该变化，它可以用程序自动地选取或者在积分之前就指定（如果一方程要积分许多次，仅仅初始数据和终结条件有小的改变，指定参数的办法也许是有用的）．这一章，我们将讨论步长和阶的选取，方法的选取将在其它各类方法讨论之后再讨论．

做这种分析的人的目的必须是用最小的工作量——人工和计算机的工作量，来达到接近于所要求的目标．我们将仅仅研究使数值计算工作达到最小的问题，但是决不能忘记为了节省几秒钟机器时间而大量耗费人力是不经济的，以及一个小的有良好性质的问题应该用最易采用和可靠的程序来积分．所要求的目标将取成在区间的终点的误差不超过指定的容许误差的条件下求问题的解．我们称它为终点问题．在某些问题中，需要在一些点上得到解．这些问题可以处理成一系列终点问题．

在很少的一类问题中，保证误差以某个数为界是可能的．它们是这样的问题，即关于误差中出现的导数，我们能够得到适当的事先估计并且能够估计 Lipschitz 常数的界．但是，一般说来，我们仅仅能够希望逼近误差和尽量去减小这逼近的工作．有两个有待研究的问题，即选取最优的步长和阶的配合以及估计出现误差公式中的导数（根据所计算的结果）．我

们将涉及在求解时估计这些量的技术，因为这些技术在自动积分中是需要的.

这本书不研究的另外一个方面，是对给出的解的误差估计或界的后天计算.在积分时通过计算(4.18)或(4.24)的办法能够做到这点，前者需要一计算过程来得到一些偏导数的最大值.为此，函数 f 必须是代数上已知的. 后者要求积分二倍数目的方程.但是，当原来的微分方程在很高的精度下积分时，(4.23)有时只要在低精度内求解，目的是检验误差是否出现迅速增长. 另一个得到严格误差界的办法是区间分析法[1). 在这个办法中，构造一个区间，保证它们包含解. 它要求用一区间函数 $\mathbf{F}(\mathbf{Y}, T)$ 代替函数 $\mathbf{f}(\mathbf{y}, t)$，使得若 $\mathbf{y}$ 和 t 在区间 $\mathbf{Y}$ 和 T 中，则函数 $\mathbf{f}(\mathbf{y}, t)$ 在区间 $F(Y, T)$ 中. 找到这样一个区间函数常常不是简单的事情，并且这种方法经常产生很坏的误差界(即大区间). 因此，它通常仅对很小一类问题适合.

5.1. 阶 的 选 取

假设我们希望在固定的范围 $[0,b]$ 上用固定阶的方法积分一问题. 更进一步，假设在全部方法中有一些方法，它们的阶为 $r=1, 2, \cdots$，并且预先利用 h_r 和 $\theta_r(t)$ 告诉每个方法的步长的最佳选取. 所谓"最佳"，意指如果在 r 阶方法中用

$$h_n = h_r\theta_r(t_n) \tag{5.1}$$

选取 h_n，则对于给定的误差范围来说，计算的工作量最小. 从定理 4.5 对于变步长的推广，有

$$\mathbf{e}_N = \boldsymbol{\delta}_r(t_N)h_r^r + O(h_r^{r+1}), \tag{5.2}$$

其中 $\boldsymbol{\delta}_r(t_N)$ 是(4.29)的解. h_r 的大小显然依赖于允许的误差

1) 见 Moore (1966), p. 90.

范围 E. 现在假设 $\theta_r(t)$ 与 E 无关[下一节将指出,这真确到 $\theta(h)$]. 在 (5.2) 中忽略 $O(h_r^{r+1})$ 项并利用

$$E \geqslant \|\mathbf{e}_N\| \cong \|\delta_r(t_N)\| h_r^r, \tag{5.3}$$

上式看作是决定 h_r 的方程.

我们的目的是确定最佳阶的方法. 假设在 r 阶方法中每步工作量是 k_r(通常必须计算导数 f 的次数,因此对于 Runge-Kutta 方法 $k_r = r$, $r \leqslant 4$.), 则在区间 $[0, b]$ 上, 整个工作量是 Nk_r, 其中 $t_N = b$, 步数 N 是 $O\left(\frac{1}{h_r}\right)$[1]. 因此整个工作量 W_r 近似等于 L_r/h_r, 其中 L_r 对所给定的问题是常数. 图 5.1 画出了不同阶方法 (5.3) 的误差 $\|\mathbf{e}_N\|$ 相对于工作量的倒数 $W_r^{-1} = h_r/L_r$ 的曲线. 曲线相互交叉的点依赖于 $\|\delta_r(t_N)\|$ 和 L_r, 它们随方程不同而改变. 但是在图 5.1 中我们能够看到两个重要

---

1) 证明如下: 设 t 是诸节点 t_n 之一, 定义 $n(t, h_r)$ 是使用 (5.1) 的法则从 0 到达 t 需要的步数. 规定 $n(t, h_r)$ 在两节点之间是线性的, 从 (5.1) 有

$$n(t_q + \theta_r(t_q)h_r, h_r) = n(t_{q+1}, h_r) = q + 1 = n(t_q, h_r) + 1$$

或

$$\frac{n(t_q + \theta_r(t_q)h_r, h_r) - n(t_q, h_r)}{h_r\theta_r(t_q)} = \frac{1}{h_r\theta_r(t_q)}$$

因此

$$\frac{\partial n}{\partial t}(t, h_r) = \frac{1}{h_r\theta_r(t_q)},$$

其中 t_q 使得 $t \in (t_q, t_{q+1})$. 于是

$$N = n(b, h_r) = \int_0^b \frac{\partial n}{\partial t}(t, h_r)dt = \int_0^b \frac{dt}{h_r\theta_r(t_q)}$$

$$= \frac{1}{h_r}\int_0^b \frac{dt}{\theta_r(t)} + \frac{1}{R_r}\int_0^b \left\{\frac{1}{\theta_r(t_q)} - \frac{1}{\theta_r(t)}\right\} dt.$$

在 θ_r 有连续一阶导数且不等于零, 同时以 Δ 为下界的假设下, 最后一项的界如下:

$$\frac{1}{R_r}\int_0^b \frac{\max|\theta'|\max|t_q - t|}{\Delta^2} dt = \frac{\max|\theta'|}{\Delta^2} b.$$

因此

$$N = \frac{1}{h_r}\int_0^b \frac{dt}{\theta_r(t)} + O(1) \qquad \text{当 } h_r \to 0. \tag{5.4}$$

的特性，即对充分小的 $\|\mathbf{e}_N\|$，W_r^{-1} 的最大值（由是最佳值）将出现在最高阶方法中．同样，对 $\|\mathbf{e}_N\|$ 充分大的值，最佳 W_r^{-1} 将出现在 $r=1$．后面这种特性可以改变，因为对于充分大的 h，在 (5.2) 中忽略 $O(h^{r+1})$ 项不合理．但是，实际问题的经验指出，低精度问题最好用低阶方法来处理．

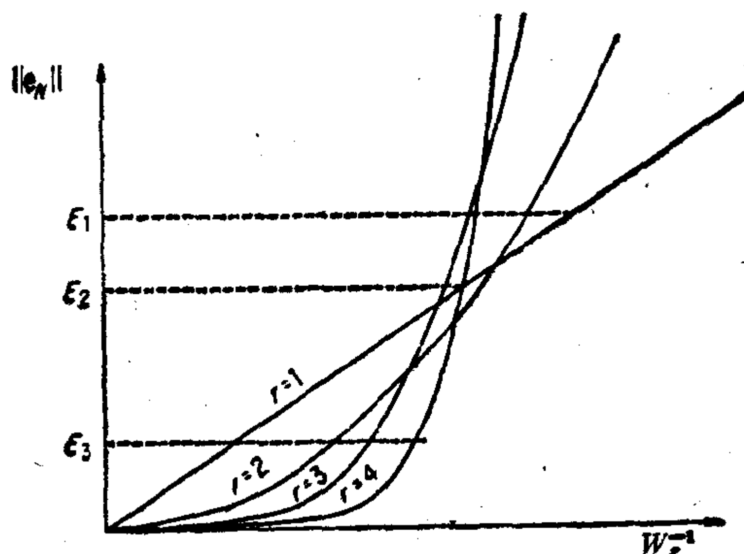


图 5.1．各种不同阶的方法误差对工作量的倒数关系

图 5.1 还可用来消除某些通常的偏见．例如，对于给定的步长（或 W_r^{-1}），并不是阶愈高的方法得到的精度愈高．图中说明，得到误差 E_1 的最经济的方法是一阶方法，而得到 E_2 是用二阶方法，得到 E_3 是用四阶方法．

我们假设阶在整个区间保持常数．若情况不是这样，则对阶是常数的任何子区间，进行同样的考虑，直到子区间仅需积分一步为止．由此我们可以看到，使用每单位步长给出最小工作量的阶方法是较好的选取．这个结论依赖于给定步上较小误差所给出的解的较小误差的假设．如果因为截断误差符号的改变使后来的误差抵消了前面的误差，这个结论也可以不真确．

为了研究每一步阶的选取，我们从(4.19)开始．假设对所有 n，$\|\mathbf{e}_n\| \leqslant E$ 且当 $E \to 0$ 时定义 $T(h, t)$ 为 $\mathbf{d}_n(h)/h$，使得 $\|T_n(h)\| = O(E)$．我们还要假设 $\boldsymbol{\phi}$ 具有连续而有界的二阶导数，于是有

$$\mathbf{e}_{n+1} = \mathbf{e}_n + h_n[\boldsymbol{\phi}_y(\mathbf{y}(t_n), t_n, 0)\mathbf{e}_n + T(h_n, t_n)] + O(E^2h_n + Eh_n^2).$$

因为 $\boldsymbol{\phi}(\mathbf{y}, t, 0) = \mathbf{f}(\mathbf{y}, t)$ 和 $h_n = h\theta(t_n) \leqslant h$，所以，这是应用 Euler 方法求得方程

$$\mathbf{e}'(t) = \mathbf{f}_y\mathbf{e}(t) + T(h\theta(t), t), \qquad \mathbf{e}(0) = 0 \tag{5.5}$$

的解，它具有附加误差 $O(E^2 + Eh)$．因为在 Euler 解中误差是 $O(h\|e''(t)\|) = O(RE)$，所以 $\mathbf{e}_n = \mathbf{e}(t_n) + O(E^2 + hE)$．

(5.5) 的解由

$$\mathbf{e}(t) = \int_0^t G(\tau, t)\mathbf{T}(\theta(t)h, \tau)d\tau \tag{5.6}$$

给出，其中 $G(\tau, t)$ 是一矩阵，满足(5.5)的齐次式，即

$$\frac{\partial G}{\partial t}(\tau, t) = \mathbf{f}_yG(\tau, t), \qquad G(\tau, \tau) = I, \tag{5.7}$$

注意 G 与 θ, h 和 $\mathbf{T}$ 无关．如果 $G(\tau, t)$ 和 $T(\theta(\tau)h, \tau)$ 的所有分量在整个区间有相同的符号，则减少 T 的方法的阶将减少整个误差 $\|\mathbf{e}(t_N)\|$．

将(5.6)改写成

$$\begin{aligned}\mathbf{e}(t_N) &= \sum_{n=0}^{N-1}\int_{t_n}^{t_{n+1}} G(\tau, t)\mathbf{T}(\theta(\tau)h, \tau)d\tau \\ &= \sum_{n=0}^{N-1} G(t_n, t_N)\mathbf{T}(\theta(t_n)h, t_n)h\theta(t_n) + O(Eh) \\ &= \sum_{n=0}^{N-1} G(t_n, t_N)\boldsymbol{d}_n(\theta(t_n)h) + O(Eh)\end{aligned} \tag{5.8}$$

也是有用的．从这里我们看到，$G(t_n, t_N)$ 是当积分进行到 t_N 时第 n 步产生的误差的放大系数．

5.2. 步长的选取

前面已经假设给出了步长函数 $\theta(t)$，现在假设知道了区间每一点供使用的最好的阶．当已知 $\theta(t)$ 时，$T(\theta(t)h, t)$ 是知道的．我们希望使工作量达到极小，条件是$\|\mathbf{e}_N\|$不超过 E．假设在点 t 对于所使用的步长的每步工作量是 $k(t)$，应用 (5.4) 到每一个有固定阶的子区间，整个工作量为

$$W = \frac{1}{h}\int_0^b \frac{k(t)}{\theta(t)}\,dt + O(1). \tag{5.9}$$

从 (5.6)

$$\|\mathbf{e}_N\| = \int_0^b \|G(t,b)\|\|\mathbf{T}(\theta(t)h,\, t)\|dt,$$

因此在服从边条件

$$E = \int_0^b \|G(t,\, b)\|\|T(\theta(t)h,\, t)\|dt \tag{5.10}$$

的要求下极小化 (5.9) [在被积函数具有不变符号的单个方程情况下，不需要横并且确信得到了真正的极小值).

应用变分法和 Lagrange 待定系数[1]的标准技术，这个问题的解由

$$\|G(t,\, b)\|\frac{\partial}{\partial\theta}\|\mathbf{T}(\theta(t)h,\, t)\| = \frac{\lambda k(t)}{\theta^2(t)} \tag{5.11}$$

给出.

---

1) 我们正在寻找一函数 $\tilde{\theta}(t)$ 极小化 (5.9)，因此，任何其他函数 $\theta(t) = \varepsilon\eta(t) + \tilde{\theta}(t)$ 在 $\varepsilon = 0$ 将有极小值．因而

$$\frac{dW}{d\varepsilon} = \int_0^b \frac{k(t)}{\tilde{\theta}^2(t)}\eta(t)dt = 0. \tag{5.12}$$

但是 (5.10) 必须对于这个 $\varepsilon \approx 0$ 成立，故 $\eta(t)$ 必须使得

$$0 = \frac{dE}{d\varepsilon} = \int_0^b \|G(t,\, b)\|\frac{\partial}{\partial\theta}\|\mathbf{T}(\tilde{\theta}(t)h,\, t)\|\eta(t)dt. \tag{5.13}$$

假如一个被积函数是另一个的常数倍，对于任何满足 (5.13) 的 $\eta(t)$，(5.13) 将推出 (5.12)．这给出方程 (5.11)．更一般的讨论可在 Courant (1950) pp. 188 和 498 中找到．

带因子的截断误差将用它的 Taylor 级数展开式中第一个没有抵消的项来近似，所以，有

$$\left\|\frac{\mathbf{d}_n(\theta h)}{\theta h}\right\| = \|\mathbf{T}(\theta h, t_n)\| = \theta^{r(t)} h^{r(t)} \|\phi(y, t_n)\| + O(h^{r+1}),$$

其中 $r(t)$ 是在点 t 所用的阶．因此，(5.11) 推出

$$r(t)\theta^{r(t)-1} h^{r(t)} \|G(t, b\| \|\phi(y, t)\| \cong \frac{\lambda k(t)}{\theta^2(t)}$$

或

$$\|G(t_n, b)\| \|\mathbf{d}_n(\theta(t_n)h\| \cong \frac{h\lambda k(t_n)}{r(t_n)}. \tag{5.14}$$

由 (5.8) 得到：当第 n 步误差对于区间端点误差的影响等于 $h\lambda k/r$ 时，得到了满足 (5.10) 并使工作量达到极小的步长的选取．常数 λ 必须满足 (5.10)．

对于阶固定在 r 的方法，由 (5.14) 推出

$$\theta^{r+1}(t_n) = \lambda h^{-r} \frac{k}{r} \frac{1}{\|G(t_n, b)\|}.$$

如果 θ 已正则化，使其最大值是 1，则可看到：θ 与所要求的误差界 E 是无关的，E 由调整 h 得到．

不巧，这个选取 θ 和 r 的规定是不实际的，因为函数 $G(t, b)$ 通常不是预先知道的．直到积分完成之前，不可能计算误差在积分区间的末端将有何种影响．下一节讨论某些实际办法，它们也没有给出最优方法，而是想克服这个缺点．首先，我们应用这些结果于某些例子．

例 1.

$$\mathbf{y}' = \mathbf{f}(t).$$

因为 $\mathbf{f}$ 不依赖于 $\mathbf{y}$，$G(b, t) = I$，所以，(5.14) 要求

$$\|\mathbf{d}(\theta(t)h)\| = \frac{h\lambda k(t)}{r(t)}, \tag{5.15}$$

$k(t)$ 依赖于阶 $r(t)$．如果应用 $r \leqslant 4$ 的 Runge-Kutta 方法，那

么 $k(t)=r(t)$. 因此，最好的策略是在每一步产生大小相同的误差，并且一步一步地选取阶，使每单位步的工作量尽可能小。

例 2.

$$y'=3t^2.$$

这是例 1 的特殊情形. 假设考虑梯形公式或 Euler 公式($r=2$ 或 1)，工作量在两方法中是一样的，故取 $k(t)=1$. 对于 Euler 方法，由 (5.15) 得到

$$\|d(\theta(t)h)\|=3t\theta^2(t)h^2=\lambda h.$$

而对于梯形公式，则为

$$\|d(\theta(t)h)\|=\frac{1}{2}\theta^3(t)h^3=\frac{\lambda h}{2}.$$

因为工作量与 $\int dt/\theta$ 成正比，我们希望使 θ 达到极大值，所以，选取

$$\theta=\max\left\{\left[\frac{\lambda}{3ht}\right]^{1/2},\ \left[\frac{\lambda}{h^2}\right]^{1/3}\right\} \tag{5.16}$$

且依其相应项是最大来选取 Euler 公式或者梯形公式. 我们可以取 $h=1$，因为它对于 θ 仅仅是一个比例尺. 因此，对于 $t\leqslant t_0$，其中 $t_0^3=\lambda/27$，应采用 Euler 公式. 对于更大的 t，梯形公式是更好的. 在区间 $[a,b]$ 上全体误差近似等于

$$\int_a^b\frac{\lambda}{\theta(t)}dt$$

$$=\begin{cases}2(b^{3/2}-a^{3/2})\sqrt{\dfrac{\lambda}{3}} & \text{如果 } b\leqslant t_0,\\ 2(t_0^{3/2}-a^{3/2})\sqrt{\dfrac{\lambda}{3}}+(b-t_0)\lambda^{2/3} & \text{如果 } a<t_0<b,\\ (b-a)\lambda^{2/3} & \text{如果 } t_0\leqslant a,\end{cases}$$

其中 $\theta(t)=\left[\dfrac{\lambda}{3t}\right]^{1/2}$. 如果允许误差大，则可使用大的 λ. 所

以，t_0 很大而且 Euler 方法效果好．当误差减小，λ 必须减小，从而在区间端点近处使用梯形公式是有利的．为了得到足够小的误差，在整个区间上应使用梯形公式．

在这个简单的例子中，也可以看出，选取 λ 来得到一个特别的误差界，也不是很简单的．

例 3.

$$y' = \alpha y, \qquad y(0) = 1.$$

对于这个问题，$G(\tau, t) = e^{\alpha(t-\tau)}$．所以，从 (5.14) 应选取 $\theta(t)$ 使得

$$e^{\alpha(b-t_n)}\|d_n(\theta(t_n)h)\| \cong \frac{h\lambda k(t_n)}{r(t)}.$$

如果使用不变的阶，则右端是常量，从而误差 d_n 应与解 $e^{\alpha t}$ 成正比．因此，不变的相对误差是最好的办法．

5.3. 误差的实际控制

前面两节利用渐近近似规定了阶和步长的最优选取，但是我们看到这些规定是不太实用的．在典型的问题中，是积分一组方程，在最后结果里产生准确的给定数目的有效数字．关于 $G(\tau, t)$ 甚至 $\partial \mathbf{f}/\partial \mathbf{y}$，在开始时是不知道的．

为了把问题简化，我们假设 $\|G(\tau, t)\| < 1$，使得误差不增大．那么，全体误差是以每一步误差的和为界的．对于这个界的最优解是置这些误差为 $\lambda k(t)/r(t)$．如果 λ 的一个值已知，每一步均可用这个公式选取，使得当到达积分区间末端时，可用最优的途径达到所产生的误差界．但是，这不是我们所需要的．我们的问题是既不知道后面诸步将取什么样的 $r(t)$ 值，也不知道使用多少步数，何况由这些来决定 λ．

$r(t)$ 的问题可假设 $k(t) = r(t)$ 来处理．对于许多方法，这是一个合适的近似．现在必须在每一步给出误差 λ，虽然

我们仍将利用 $K(t)$ 的最好近似来选取每单位步长，给出最小工作量的阶．按这个规定，全体误差是 $N\lambda$，而 N 仍然是未知的．这个问题的解决办法是控制在每一步中的误差小于 $\lambda h\theta$，这就是使每单位步长的误差不变，而每单位步的工作量达到极小．这样，在长度为 b 的区间上，全体误差是 λb．因此，λ 可取为 E/b．这样得到的选取不是最优的，但差别不太大，且得到一个误差大小合理的答案．[我们必须记住，与 $O(E^2h+Eh^2)$ 相同的项在误差推导中被忽略，所以，推导将不保证所要求的精度．]

如果 $G(\tau, t)$ 不是不变的又如何呢？一般地说，勿需讨论，但是对于相当大的一类问题，我们阐明一个类似于上面方法的技术．考虑形如

$$\mathbf{y}' = A\mathbf{y} + \mathbf{g}(t) \tag{5.17}$$

的线性方程．我们有

$$G(\tau, t) = \exp\{A(t-\tau)\}$$

的解是

$$\mathbf{y}(t) = \int_0^t G(\tau, t)\mathbf{g}(\tau)d\tau + G(0, t)\mathbf{y}(0). \tag{5.18}$$

如果 $\mathbf{g}$ 和 G 的分量有少量相互抵消且 $\mathbf{y}(0)$ 使得 $\mathbf{y}(t)$ 作类似于第二项的变化，我们可以应用这个解来控制误差．这就是在每一步使误差等于 $Eh\theta\|\mathbf{y}(t)\|/b$．在 5.2 中曾指出，误差应使得它们对最后误差的影响相等，因此 $\|G(t_n, t_N)d_n(\theta\times(t_n)h\| = \|G(t_{n+1}, t_N)d_{n+1}(\theta(t_{n+1})h\|$．对这个问题，

$$G(a, c) = G(a, b)G(b, c) = G(b, c)G(a, b),$$

所以，我们可改而要求

$$\|G(t_n, t_{n+1})d_n(\theta(t_n)h)\| = \|d_{n+1}(\theta(t_{n+1})h)\|. \tag{5.19}$$

这就是第 n 步产生的误差对 $n+1$ 步全体误差的影响和第 $n+1$ 步产生的误差同样大小，根据假设，(5.18) 中 $\mathbf{y}(t)$ 的

性质类似于最后一项，我们看到

$$\|G(t_n, t_{n+1})\| \cong \|\mathbf{y}(t_{n+1})\| / \|\mathbf{y}(t_n)\|.$$

因此，如果使截断误差与 $\|\mathbf{y}(t)\|$ 成正比，(5.19) 就近似地被满足. 通过引入的附加因子 $Eh\theta/b$，将有一个相对于最后答案的近似等于 E 的全体误差. 因此，如果 E 是 10^{-k}，则希望得到 k 位有效数字. 通过代入 (5.8) 得到

$$\begin{aligned}\|\mathbf{e}(b)\| &\leqslant \sum_{n=0}^{N-1} \|G(t_n, b)\| \frac{Eh\theta(t_n)\|y(t_n)\|}{b} + O(Eh) \\ &\cong \sum_{n=0}^{n-1} \|G(t_n, b)\mathbf{y}(t_n)\| \frac{Eh\theta(t_n)}{b} \\ &\cong \sum_{n=0}^{n-1} \|\mathbf{y}(b)\| \frac{Eh\theta(t_n)}{b} = \|\mathbf{y}(b)\| E.\end{aligned}$$

必须强调，这些是很粗糙的近似且没有给出任何精度的保证. 构造一个实际误差比 E 任意小或者大的例子是简单的事情. 例如考虑

$$y' = \lambda(y - e^t) + e^t, \qquad y(0) = 1,$$

它有解 $y = e^t$，因此误差将相对于 e^t 来控制. 然而 $G(\tau, t) = e^{\lambda(t-\tau)}$ 且误差应相对于它来控制. 如果 $\lambda \geqslant 0$，那么在 t_N 实际误差将比 E 大得多；反之，如果 $\lambda < 0$，误差是小的.

5.4. 局部截断误差的估计

对于步长和阶控制所得到的准则假设了局部截断误差大小的知识. 在一通用的程序中，只需要根据数值解的知识估计局部截断误差，且应用这些信息来控制步长和阶的选取. 如果应用 Taylor 级数方法，为了决定用哪一阶，可能要计算几个高一阶的导数(但是，注意所考虑的最高阶方法的工作量在计算这些导数时已经完成). Taylor 级数方法通常是不实用的，实用的是 Runge-Kutta 方法. 它们的误差项包含 f 的偏导数

的组合，计算这些偏导数是不实际的．然而，关于 Runge-Kutta 方法的误差控制有两个通常使用的技术．

5.4.1. 步数加倍

这个技术对所有方法均适用，而通常仅限于 Runge-Kutta 方法，因为对于其他方法有更好的办法．每个长为 h 的基本步做两次，一次是步长为 $h/2$ 的两步和一次步长为 h 的一步．因为误差形如 $h^{r+1}\boldsymbol{\phi}(\mathbf{y}, t) + O(h^{r+2})$，可比较这两个结果来估计 $\|\boldsymbol{\phi}\|$．如果步长为 h 的结果是 $\mathbf{y}_2$，而步长为 $h/2$ 的两步结果是 $\mathbf{y}_1$，则有

$$\|\mathbf{y}_2 - \mathbf{y}_1\| = h^{r+1}(1 - 2^{-r})\|\boldsymbol{\phi}(t)\| + O(h^{r+2}).$$

这个仅可用来估计正在使用的某阶方法的误差．对于选取 Runge-Kutta 方法的阶，没有普遍有效的技术．

100页上的程序，是一个典型的 Runge-Kutta 子程序，每次进行二重步积分．计算每一个分量上的误差且被 $YMAX(I)$ 除．开始时可安排某些分量较之其他分量有更大的权，修正后就包含至今已被计算的 $Y(I)$ 的最大值．最坏的情况是带比例的误差被限制小于一个输入参数 EPS．计算下一步的 h 值，使它近似等于下一步带比例误差 EPS 所需要的大小．在语句标号 6 下面的语句中，因子 0.99 的作用是保持H稍小，以便如果下一步的截断误差稍大，所推荐的H也足够小．

如果带比例的误差比 EPS 大，此步舍弃，且用新推荐的步长H重做，但以 $H \geqslant HMIN$ 为条件（$HMIN$ 是子程序参数）．这个子程序近似地保持相对误差在最大模中不变．如果使误差与 hE 成正比，则所有引用的 EPS 应该用 $H*EPS$ 代替，且在语句标号为 6 的语句后面的语句中指数 -0.2 改为 -0.25．

对不同的相对误差 E，从 $y(0) = 1$ 到 $t = 20$ 积分 $y' =$

$-y$，结果在表 5.1 中列中． EPS取成 10^{-k}，同时 $YMAX(I)$ 在每一步之前定为 $Y(I)$． 在图 5.2 中画出 $t=20$ 的实际相对误差对函数求值的曲线．

表 5.1　四阶 Runge-Kutta 方法的结果

k	相对误差	函数计值数目	步数
3	0.67500D-02	187	17
4	0.21449D-02	286	26
5	0.38661D-03	440	40
6	0.62225D-04	704	64
7	0.98484D-05	1100	100
8	0.15670D-05	1738	158
9	0.24796D-06	2728	248
10	0.39216D-07	4367	397
11	0.62065D-08	6798	618

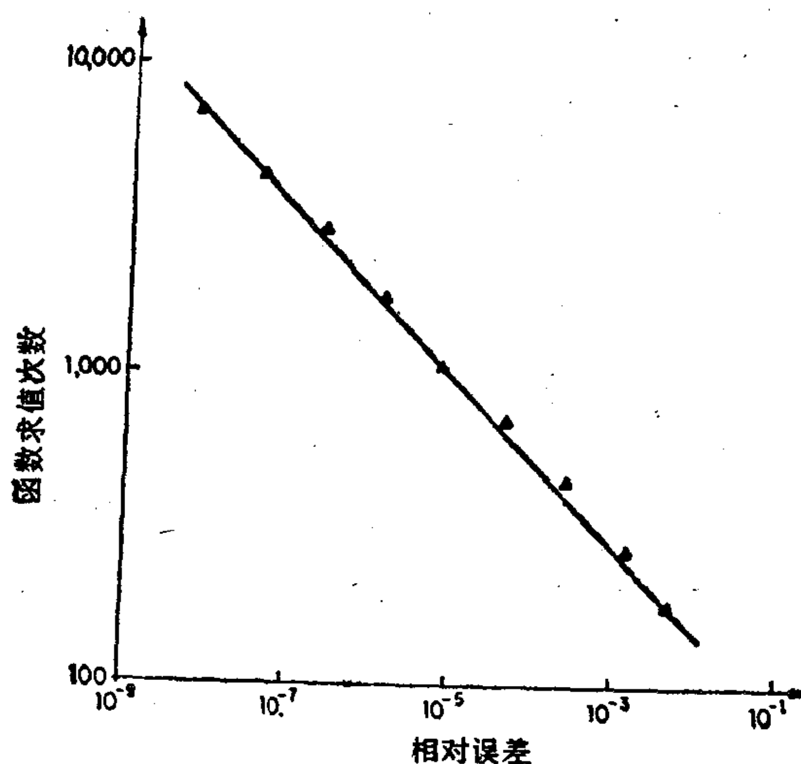


图 5.2.　四阶 Runge-Kutta 方法的结果

```
      SUBROUTINE DIFSUB (N,T,Y,DY,H,HMIN,EPS,YMAX,ERROR,KFLAG,
     1                   JSTART)
      IMPLICIT REAL *8 (A-H,O-Z)
C*********************************************************************
C*  THE PARAMETERS TO THIS INTEGRATION SUBROUTINE HAVE               *
C*  THE FOLLOWING MEANINGS..                                          *
C*    N       THE NUMBER OF FIRST ORDER DIFFERENTIAL EQUATIONS        *
C*    T       THE INDEPENDENT VARIABLE                                *
C*    Y       THE DEPENDENT VARIABLES, UP TO 10 ARE ALLOWED.          *
C*    DY      AN ARRAY OF 10 LOCATIONS WHICH WILL CONTAIN THE         *
C*              VALUES OF THE DERIVATIVES AT THE START OF THE INTERVAL.*
C*    H       THE STEP SIZE THAT SHOULD BE ATTEMPTED.  IT MAY BE      *
C*              INCREASED OR DECREASED BY THE SUBROUTINE.             *
C*    HMIN    THE MINIMUM STEP SIZE THAT SHOULD BE ALLOWED ON THIS    *
C*              STEP.                                                 *
C*    EPS     THE ERROR TEST CONSTANT.  THE ESTIMATED ERRORS ARE      *
C*              REQUIRED TO BE LESS THAN EPS*YMAX IN EACH COMPONENT.  *
C*              IF YMAX IS ORIGINALLY SET TO +1 IN EACH COMPONENT,    *
C*              THE ERROR TEST WILL BE RELATIVE FOR THOSE COMPONENTS  *
C*              GREATER THAN 1 AND ABSOLUTE FOR THE OTHERS.           *
C*    YMAX    THE MAXIMUM VALUES OF THE DEPENDENT VARIABLES ARE       *
C*              SAVED IN THIS ARRAY. IT SHOULD BE SET TO +1 BEFORE    *
C*              THE FIRST ENTRY. (SEE THE DESCRIPTION OF EPS).        *
C*    ERROR   THE ESTIMATED SINGLE STEP ERROR IN EACH COMPONENT       *
C*    KFLAG   A COMPLETION CODE WITH THE FOLLOWING MEANINGS..         *
C*                    -1   THE STEP WAS TAKEN WITH H = HMIN           *
C*                         BUT THE REQUESTED ERROR WAS NOT ACHIEVED.  *
C*                    +1   THE STEP WAS SUCCESSFUL.                   *
C*    JSTART AN INITIALIZATION INDICATOR WITH THE MEANING..           *
C*                    -1   REPEAT THE LAST STEP, RESTORING THE        *
C*                         VALUES OF Y AND YMAX THAT WERE USED        *
C*                         LAST TIME.                                 *
C*                    +1   TAKE A NEW STEP.                           *
C*********************************************************************
      DIMENSION Y(10),DY(10),YMAX(10),YSAVE(10),Y1(10),Y2(10),Y3(10)
     1          ,ERROR(10),DYN(10),YMAXSV(10)
      IF(JSTART.LT.0) GO TO 2
C*********************************************************************
C* SAVE THE VALUES OF Y AND YMAX IN CASE A RESTART IS NECESSARY.      *
C*********************************************************************
      DO 1 I = 1,N
      YSAVE(I) = Y(I)
    1 YMAXSV(I) = YMAX(I)
C*********************************************************************
C*  CALCULATE THE INITIAL DERIVATIVES.                                *
C*********************************************************************
      CALL DIFFUN(T,Y,DYN)
      GO TO 4
C*********************************************************************
C*  RESTORE THE INITIAL VALUES OF Y AND YMAX FOR A RESTART.           *
C*********************************************************************
    2 DO 3 I = 1,N
      Y(I) = YSAVE(I)
    3 YMAX(I) = YMAXSV(I)
    4 KFLAG = 1
C*********************************************************************
C*  SAVE THE FINAL VALUE OF T AND CALCULATE THE HALF STEP.            *
```

```
C*********************************************************************
 5      A = H + T
        HHALF = H*0.5D0
C*********************************************************************
C*    PERFORM ONE FULL RUNGE KUTTA STEP                              *
C*********************************************************************
        CALL RK1(N,T,YSAVE,DYN,H,Y1)
C*********************************************************************
C*    NOW PERFORM TWO HALF INTERVAL RUNGE KUTTA STEPS                *
C*********************************************************************
        CALL RK1(N,T,YSAVE,DYN,HHALF,Y2)
         THALF = T + HHALF
        CALL DIFFUN(THALF,Y2,DY)
        CALL RK1(N,THALF,Y2,DY,HHALF,Y3)
        ERRMAX = 0
C*********************************************************************
C*    CALCULATE THE NEW MAX Y'S, THE ERRORS AND THE MAY              *
C*    RELATIVE ERRORS.                                               *
C*********************************************************************
        DO 6 I = 1,N
          YMAX(I) = DMAX1(YMAX(I),DABS(Y1(I)),DABS(Y2(I)),DABS(Y3(I)))
          ERROR(I) = DABS((Y3(I) - Y1(I))/31.0D0)
          ERRMAX = DMAX1(ERRMAX,ERROR(I)/(EPS*YMAX(I)))
C*********************************************************************
C*    CALCULATE THE IMPROVED VALUE OF Y BY ELIMINATING THE           *
C*    ESTIMATED ERROR.                                               *
C*********************************************************************
          Y(I) = (32.0D0*Y3(I) -Y1(I))/31.0D0
 6        CONTINUE
        IF (ERRMAX.EQ.0) H = H*2.0D0
        IF (ERRMAX.GT.0) H = H*ERRMAX**(-0.2)*0.99
        IF (ERRMAX.GT.1.0D0) GO TO 8
        KFLAG = 1
 7      T = A
        RETURN
 8      IF (H.GT.HMIN) GO TO 5
        IF (KFLAG.LT.0) GO TO 7
        H = HMIN
        KFLAG = -1
        GO TO 5
        END
C*********************************************************************
C*    THIS SUBROUTINE PERFORMS ONE RUNGE KUTTA STEP.                 *
C*.     ARGUMENTS ARE..                                              *
C*      N     -  NUMBER OF EQUATIONS.                                *
C*      T     -  INITIAL VALUE OF INDEPENDENT VARIABLE               *
C*      Y     -  INITIAL VALUE OF DEPENDENT VARIABLES.               *
C*      DY    -  INITIAL VALUE OF DERIVATIVES.                       *
C*      H     -  STEP SIZE                                           *
C*      Y1    -  THE ANSWER IS RETURNED HERE.                        *
C*********************************************************************
        SUBROUTINE RK1(N,T,Y,DY,H,Y1)
        IMPLICIT REAL*8 (A-H,O-Z)
        DIMENSION Y(10),DY(10),Y1(10),Y2(10),Y3(10),DY1(10)
        HHALF = H*0.5D0
        DO 1 I = 1,N
          Y2(I) = Y(I) + HHALF*DY(I)
        CALL DIFFUN(T + HHALF,Y2,DY1)
        DO 2 I = 1,N
          Y3(I) = Y(I) + HHALF*DY1(I)
          Y2(I) = Y2(I) + 2*Y3(I)
        CALL DIFFUN(T + HHALF,Y3,DY1)
        DO 3 I = 1,N
          Y3(I) = Y(I) + H*DY1(I)
          Y2(I) = Y2(I) + Y3(I)
        CALL DIFFUN(T + H,Y3,DY1)
        DO 4 I = 1,N
          Y1(I) = (Y2(I) - Y(I) + HHALF*DY1(I))/3.0D0
        RETURN
        END
```

5.4.2. Runge-Kutta-Merson 方法

这个方法是四阶 Runge-Kutta 过程,同时给出单步误差的近似值. 它需要一次附加的函数求值.这样两步将有 10 次函数求值, 而在上一节中讨论的步数加倍过程是 11 次. 方程由 Merson 于 1957 年给出如下:

$$
\begin{aligned}
&\eta_0 = y_n, \qquad k_0 = hf(\eta_0) \\
&\eta_1 = \eta_0 + k_0/3, \qquad k_1 = hf(\eta_1), \\
&\eta_2 = \eta_0 + \frac{k_0 + k_1}{6}, \qquad k_2 = hf(\eta_2), \\
&\eta_3 = \eta_0 + \frac{k_0 + 3k_2}{8}, \qquad k_3 = hf(\eta_3), \qquad (5.20) \\
&\eta_4 = \eta_0 + \frac{k_0 - 3k_2 + 4k_3}{2}, \qquad k_4 = hf(\eta_4), \\
&y_{n+1} = \eta_5 = \eta_0 + \frac{k_0 + 4k_3 + k_4}{6}.
\end{aligned}
$$

仅考虑求积公式,容易证明

$$
\begin{aligned}
\eta_1 &= y\left(t_n + \frac{h}{3}\right) + O(h^2), \\
\eta_2 &= y\left(t_n + \frac{h}{3}\right) + O(h^3), \\
\eta_3 &= y\left(t_n + \frac{h}{2}\right) + O(h^4), \\
\eta_4 &= y(t_n + h) + O(h^4),
\end{aligned}
$$

最后

$$\eta_5 = y_{n+1} = y(t_{n+1}) + O(h^5).$$

例如,用梯形公式获得 η_2, 带有来自 k_1 的附加误差 $O(h^3)$, 同时用 Simpson 公式获得 η_5, 带有来自 k_3 和 k_4 的附加误差 $O(h^5)$. 如果我们研究 η_4 中的 $O(h^4)$ 误差, 则它仅仅包括

$f_{ijk}f^if^jf^k$ 和 $f_{ij}f^i_kf^jf^k$ 项. 假如 f 的所有二阶偏导数是零，也就是：如果 $f(y, t)$ 形如 $Ay + bt$，那么 $\eta_4 = y(t_{n+1}) + O(h^5)$. 在这种情况下，Merson 证明了

$$\eta_4 = y(t_{n+1}) - \frac{1}{120}h^5y^{(5)} + O(h^6) \qquad (5.21)$$

和

$$\eta_5 = y(t_{n+1}) - \frac{1}{720}h^5y^{(5)} + O(h^6). \qquad (5.22)$$

于是差 $\eta_5 - \eta_4$ 是一个局部误差的表示.在一些自动选取步长的方法中,这个方法已被成功地使用着.

问　　题

1. 对方程

   (a) $y' = y, \quad y(0) = 1,$

   (b) $y' = -y, \quad y(0) = 1$

   使用 Taylor 级数方法讨论阶和步长的最佳选取;证明常数阶方法是最优的.

2. 在 Euler 方法中对方程 $y' = 3t^2$ 精确地表示误差项且利用它来决定最优的步长分配.

3. 在问题 2 中,用你们导出的法则从 $y(0.1) = 10^{-3}$ 到 $t = 1$ 积分微分方程. 再用 5.2 和 5.3 中所推出的法则，即单步截断误差不变和单步截断误差与步长成正比，比较在这三种情形中步数对精度的图形.

4. 证明在 5.4.1 中倒数第二段的最后一句.

5. 证明等式 (5.21) 和 (5.22).

6. 重复在表5.1中说明的积分,其中局部误差界用 $EPS*H*|YMAX(I)|$ 代替 EPS. 画出结果并把它们同图 5.2 中的图形进行比较.

7. 在区间 [0,1] 上，用梯形法则积分

   $$y' = -1000(y - e^t) + e^t, \quad y(0) = 1.$$

   使用由如下条件所确定的步长控制:

(a) 相对解的局部误差保持不变；

(b) 方程(5.14).

对于几个不同的所要求的误差界重复积分并且画出误差对于函数求值次数的图形。

6. 外插方法

前面已经看到,用消去误差项的 Richardson 外插过程的办法可以改进数值解的阶. 于是,在某固定点 t, 解与步长 h 之间呈下形:

$$y(t, h) = y(t) + \sum_{i=1}^{m} \tau_i(t)h^i + O(h^{m+1}), \qquad (6.1)$$

并且想通过对 $h = h_0, h_1, \cdots, h_m$ 求 $y(t, h)$ 的值来消去 τ_1, $\tau_2, \cdots, \tau_m$, 这是自然的. 形如(6.1)的展开式存在的条件已由 Gragg (1963) 和 Stetter (1965) 研究过了. 如果这样的級数存在, 我们可以考虑用某个函数 $R_m(t, h)$ 来近似它, 这函数有 $(m+1)$ 个未知数通过要求

$$R_m(t, h_j) = y(t, h_j) \quad (j = 0, 1, \cdots, m) \qquad (6.2)$$

来确定,以便用 $R_m(t, 0)$ 近似 $y(t)$.

6.1. 多项式外插

例如, 假设 $R_m(t, h)$ 是关于 h 的 m 次多项式, 于是若(6.1)式成立,则当 h 趋于 0 时,$R_m(t,h)$ 应是 $y(t)+O(h^{m+1})$, 即有固定的 $\alpha_i(i = 0, 1, 2\cdots)$, 使$h_i = \alpha_i h$. 在这种意义下, h 是 h_i 同时趋于零的度量. 这是一简单的多项式内插. 在区间 $(0, t)$ 上对 $i = 0, 1, \cdots, m$用步长 h_i 积分微分方程得到 $y(t, h_i)$, 其中 $h_0 > h_1 > h_2 \cdots > h_m > 0$, 进行多项式外插. 然后计算通过这些结果的关于 h 的m 次多项式在 $h=0$ 的值.

通常完成它的最简单的途径是 Aitken 内插过程. 在这过程中通过 (h_j, y_j) 确定 $R_m^i(h)$ 为 h 的m 次的唯一的多项式,

$j=i, i+1, \cdots, i+m$. 因此，$R_0^i(h)=y(t, h_i)$. 显然，函数

$$P(h)=\alpha(h)R_{m-1}^i(h)+(1-\alpha(h))R_{m-1}^{i+1}(h)$$

对于 $j=i+1, \cdots, i+m-1$ 在 $h=h_j$ 取值 $y(t, h_j)$. 要求 $\alpha(h_i)=1$ 和 $\alpha(h_i+m)=0$，我们还可以使它对于 $j=i$ 和 $j=i+m$ 也成立. 如果 $\alpha(h)$ 关于 h 是线性的，$P(h)$ 的次数为 m，则 $P(h)$ 就是 $R_m^i(h)$ 具有所要求的值的唯一的 m 次多项式. 因而有

$$\alpha(h)=\frac{h-h_{i+m}}{h_i-h_{i+m}},$$

得到

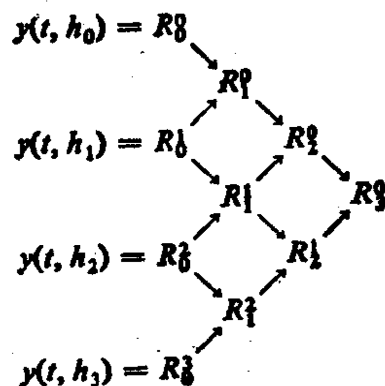
$$R_m^i(h)=\frac{1}{h_i-h_{i+m}}[(h-h_{i+m})R_{m-1}^i(h)+(h_i-h)R_{m-1}^{i+1}(h)]. \tag{6.3}$$

这提供了一个关于构造对 $R_m^i(h)$ 近似的三角形法则. 在现在的情况下，我们希望知道 $R_m^i=R_m^i(0)$ 的值. 它们可以如表 6.1 所示那样来构造，按照关系式

$$R_m^i=\frac{h_iR_{m-1}^{i+1}-h_{i+m}R_{m-1}^i}{h_i-h_{i+m}}=R_{m-1}^{i+1}+\frac{R_{m-1}^{i+1}-R_{m-1}^i}{(h_i/h_{i+m})-1} \tag{6.4}$$

用两个值去构造另一个近似值（如表中所示）. 如果这个方法应用于一个一阶方法的解，如 Euler 方法，那么值 R_m^i 当 i 或者 m 两者都增长时，将构成对解的较好的近似. 事实上可以证明，如果适当的可微条件成立，R_m^i 的误差将以形如 $M_mh_i^m$ 的某个量为界. 这样，当 $i\to\infty$ 使得 $h_i\to 0$ 时，有 $R_m^i\to y(t,0)$. 由于 M_m 将依赖于解的导数和求解的方法，对于固定的 i，如果没有进一步的限制，当 $m\to\infty$ 时，还不能肯定是否收敛. 但是，对于固定的 i，当 m 增加时，如果近似过程不收敛，则用保持 m 不变和加长 i 的办法，可以有效地减小初始步长.

表 6.1 AITKEN 杆值



6.1.1. 多项式外插的例

这个过程的一个例子在下面的表 6.2 中给出.用步长 $h=2^i$, $i=0, 1, \cdots, 9$,重新积分方程 $y'=-y$, $y(0)=1$. 相应的误差在表 6.3 中给出. 每一列以 h 的幂次增加的方式收敛,因此,解的第四列按 h^4 收敛.这点在表 6.4 中说明,它由表 6.3 的误差除以 h_i^m 构成.

6.1.2. 舍入误差的影响

我们看到这种类型的方法可以有任意大的阶（通过使 m 很大的办法)并且能够减小步长,如果还利用较大步的计算结果的话(对固定的 m 增大 i). 实际上,这个过程受舍入误差和不稳定性所限制.增加 m 或 i 时,在用 Euler 方法的基本积分中，运算次数以 2^{m+i} 的速度增加. 这意思是: 如果在每一步出现最坏的舍入误差，则解对每一个 i 或 m 的增大将失去一个 2 进位精度. 稍微克服这个问题的一个办法是应用选取更小的步长的不同办法. 在前面的例子中，对每一个外插的增加行，我们把步长减半. 不这样，假设我们每次乘以数量 $1/q(1<q<2)$ 来近似地减小步长在 R_0^i 的最坏情况的舍入误差,将与 q 成比例,因为它大约等于步数. 从(6.4)看到,关

表 6.2　对 $y' = -y,\ y(0) = 1$ 的 R^i_m

i \ m	0	1	2	3	4	5	6
0	0.0	0.500000000	0.343750000	0.370105743	0.367774922	0.367881947	0.367879410
1	0.250000000	0.382812500	0.366811275	0.367920598	0.367878603	0.367879450	0.367879441
2	0.316406250	0.370811582	0.367781933	0.367881228	0.367879424	0.367879441	0.367879441
3	0.343608916	0.368539345	0.367868816	0.367879536	0.367879441	0.367879441	0.367879441
4	0.356074130	0.368036448	0.367878196	0.367879447	0.367879441	0.367879441	
5	0.362055289	0.367917759	0.367879290	0.367879442	0.367879441		
6	0.364986524	0.367888908	0.367879423	0.367879441			
7	0.366437716	0.367881794	0.367879439				
8	0.367159755	0.367880028					
9	0.367519891						

表 6.3　R^i_m 的误差

i \ m	0	1	2	3	4	5	6
0	−0.367879441	0.132120559	−0.024129441	0.002226302	−0.000104519	0.000002506	−0.000000031
1	−0.117879441	0.014933059	−0.001068166	0.000041157	−0.000000838	0.000000009	−0.0
2	−0.051473191	0.002932140	−0.000097508	0.000001786	−0.000000018	0.0	−0.0
3	−0.024270525	0.000659904	−0.000010625	0.000000095	−0.0	0.0	−0.0
4	−0.011805311	0.000157007	−0.000001245	0.000000006	−0.0	0.0	
5	−0.005824152	0.000038318	−0.000000151	0.0	−0.0		
6	−0.002892917	0.000009466	−0.000000019	0.0			
7	−0.001441725	0.000002353	−0.000000002				
8	−0.000719686	0.000000586					
9	−0.000359550						

表 6.4　R^i_m 的误差除以 h_i^m

i \ m	0	1	2	3	4	5	6
0	−0.367879441	0.132120559	−0.024129441	0.002226302	−0.000104519	0.000002506	−0.000000031
1	−0.235758882	0.059732235	−0.008545326	0.000658514	−0.000026827	0.000000572	−0.000000006
2	−0.205892765	0.046914247	−0.006240528	0.000457311	−0.000017967	0.000000372	−0.000000004
3	−0.194164203	0.042233851	−0.005440234	0.000389934	−0.000015080	0.000000309	−0.000000007
4	−0.188884972	0.040193764	−0.005099041	0.000361659	−0.000013883	0.000000254	
5	−0.186372861	0.039237693	−0.004940815	0.000348643	−0.000013391		
6	−0.185146683	0.038774492	−0.004864550	0.000342366			
7	−0.184540832	0.038546466	−0.004827103				
8	−0.184239688	0.038433331					
9	−0.184089557						

于 R_1^i 下一列的误差可以达到

$$q^{i+1}+\frac{q^{i+1}+q^i}{q-1}=q^{i+1}\left(\frac{q+q^{-1}}{q-1}\right).$$

在下一步，R_2^i 的误差可以大如

$$\frac{q+q^{-1}}{q-1}\left[q^{i+2}+\frac{q^{i+2}+q^{i+1}}{q^2-1}\right]=q^{i+2}\frac{q+q^{-1}}{q-1}\cdot\frac{q^2+q^{-1}}{q^2-1}$$

以及 R_m^i 的误差可以达到

$$q^{i+m}\frac{q+q^{-1}}{q-1}\cdot\frac{q^2+q^{-1}}{q^2-1}\cdots\frac{q^m+q^{-1}}{q^m-1}. \qquad (6.5)$$

的确，让 q 接近 1，q^{i+m} 项的影响可以减少，但是表达式(6.5)的第二部分随着 $q\to 1$ 时迅速增大.

6.1.3. 稳定性

这一章所要讨论的全部方法的稳定性问题是一个未解决的问题. 经验证明，在把基本的步长 h_0 减小到使得 $|h_0\partial f/\partial y|$ 接近 1 以前具有负得很大的 $\partial f/\partial y$ 值的方程引起严重的不稳定性问题，而这是一个需要多多注意的地方.

6.1.4. 高阶方法

前面多项式插值中所用的基本方法是 Euler 方法，即一个一阶方法. 例如，为了得到四阶方法，必须进行三次外插. 如果我们用序列步长 $h, h/2, h/4, h/8, \cdots$, 导函数 f 的计算总共需要 12 次，相比之下，Runge-Kutta 四阶方法是四次. 利用序列 $h,h/2,h/3,h/4$ (这是最少的，因为步长必须不同且为基本步长 h 的约量)，我们可以把函数计算的次数减少到七次，象在上一节所讨论的那样，虽然可能增加舍入误差. 另外一个办法是用高阶方法开始，例如，如果使用一个 r 阶方法，当应用适当的外插公式时，近似值 R_m^i 将有误差 $O(h^{r+m})$. 如果可采用形如

$$y_h(t) = y(t) + \sum_{i=1}^{\infty} \tau_i h^{ri}$$

的数值积分公式，即积分是 h^r 的函数，则外插公式在每一步可增加阶 r，因此 R^i_m 将有误差 $O(h^{r(m+1)})$. 一种这样的公式对 $r = 2$ 是已知的，它有下面的形式：定义

$$t_n = nh,$$

$$\eta(0,h) = y(0); \qquad (6.6a)$$

$$\eta(t_1, h) = y(0) + hf(y_0, 0); \qquad (6.6b)$$

$$\eta(t_{n+1}, h) = \eta(t_{n-1}, h) + 2hf(\eta(t_n, h), t_n)$$

$$n = 1, 2, \cdots, N-1, \quad \text{其中 } t_N = t; \qquad (6.6c)$$

$$y(t, h) = \frac{1}{2}[\eta(t_{N-1}, h) + \eta(t_N, h) + hf(\eta(t_N, h), t_N)]. \qquad (6.7)$$

Gragg (1965) 已证明，倘若步数总是偶的或者总是奇的，以及以 $y(t)$ 是充分可微为前提，在这个方法中所定义的$y(t, h)$ 是 h 的偶次幂级数. Bulirsch 和 Stoer (1966) 推荐用全偶的序列，以 $h_0 = H/2$ 开始，其中H是所要积分的基本区间的长度.

从 $t = 0$ 开始，在每一个长度为H的区间上积分为一个新的初值问题.外插过程应用于第一个区间的结果 $y(H, h_i)$，对于所要求的精度得到 $y(H)$. 从这儿开始进行在区间 $[H, 2H]$ 上的外插过程，得到 $y(2H)$，如此等等.

这个过程的一个例子表示在表 6.5 和表 6.6 之中.所使用的 h 值的序列是 $H/2, H/4, H/6, H/8, H/12, H/16, \cdots$.所用的外插公式类似于 (6.4)，其中用 $(h_i/h_{i+m})^2$ 代替 h_i/h_{i+m}. 再一次从 0 到 1 积分 $y' = -y$，对于 $y(0) = 1$ 和 $H = 1$. R^i_m 的误差的 10^9 倍表示在表 6.5 中. 可以看到，它们比在表 6.3 中的数字要少得多. 两表中函数计算的次数不是一种简单的

关系，所以表 6.6 表示表 6.3 函数求值的次数除以表 6.5 函数求值的次数. 如所期望的，对高精度来说，表 6.5 中所要求的计算量是比较小的.

表 6.5. 对多项式外插 R_m^i 的误差

i	h_i	m: 0	1	2	3	4	5
0	$\frac{1}{2}$	7 120 559	1 930 247	−287 188	32	0	0
1	$\frac{1}{4}$	3 214 309	276 508	−9 062	1	0	
2	$\frac{1}{6}$	1 576 434	73 481	−907	0		
3	$\frac{1}{8}$	817 384	8 893	−122			
4	$\frac{1}{12}$	417 681	4 800				
5	$\frac{1}{16}$	236 932					

表 6.6. 表 6.3 函数求值的次数除以表 6.5 函数求值的次数

i \ m	0	1	2	3	4	5
0	$\frac{1}{3}$	$\frac{2}{7}$	$\frac{5}{13}$	$\frac{12}{21}$	$\frac{27}{33}$	$\frac{58}{49}$
1	$\frac{2}{5}$	$\frac{5}{11}$	$\frac{12}{19}$	$\frac{27}{31}$	$\frac{58}{47}$	
2	$\frac{4}{7}$	$\frac{11}{15}$	$\frac{26}{27}$	$\frac{57}{43}$		
3	$\frac{8}{9}$	$\frac{23}{21}$	$\frac{54}{37}$			
4	$\frac{16}{13}$	$\frac{47}{29}$				
5	$\frac{32}{17}$					

6.2. 有理函数外插

对于许多应用来说，有理函数插值比多项式插值更精确. 一个有理函数是两个多项式的商. 我们将限于研究形如

$$R_m(h) = \frac{P_m(h)}{Q_m(h)} = \frac{P_0 + P_1 h + \cdots + P_\mu h^\mu}{q_0 + q_1 h + \cdots + q_\nu h^\nu} \qquad (6.8)$$

的有理函数，其中 $\mu = [m/2]$（即 $m/2$ 的整数部分），$\nu = m - n = [(m+1)/2]$（这被称为对角有理多项式）. 如果用 (6.8)

代替上节多项式近似，则得到了 Bulirsch-Stoer (1966) 算法．在这方法中，对步长序列 $H/2,\ H/4,\ H/6,\ H/8,\ H/12,\ \cdots$，以 (6.6)所描述的二阶积分法则来积分方程，得到我们叫做 R_0^0, R_0^1, R_0^2, $\cdots$等等的结果．然后，以某种类似于对多项式的方程 (6.4) 的方式将这些数结合起来以得到有理插值的结果．代替方程 (6.4) 所用的算术关系复杂得多，所以把它做成引起尽可能少的舍入误差的形式是重要的．

随着计算 $y(t,h_m)=R_0^m$ 的每一个新值，就可计算 R_k^{m-k}的值，$k=1,2,\cdots,m$．当两次逐次近似 R_k^{m-k} 和 R_k^{m-k+1} 很接近时，过程停止．至此它们的差用来做为还存在的误差量的表示．直到使用小的 h_m，如果还必须继续这个过程，它表示原来步长H的值太大． Bulirsch 和 Stoer 建议用继续该过程来控制它直到m 为 6．如果它在误差范围内已经收敛，那么就可以采纳这个结果，而如果它对 $m<6$ 已经收敛，则为了节省工作量可增大基本步长H(但是注意，也许必须限制H使得在所要求的打印点上获得计算结果)．如果方法在m 为 6 的时间不收敛，那么逐次计算 R_6^{m-6} 的值直到收敛．如果m变得太大，则必须减小基本步长H，因为在较小的 h_i 子步中的舍入误差积累也许否定外插的假设，同时在大的子步中稳定性问题也许妨碍外插的收敛性．一典型的程序能够继续到 R_6^{10}，如果差尚未小，则减小基本步长H并且重新开始．

我们首先研究有理函数插值和推导一些递推关系式．假设有 (6.8) 型的公式使得 $R_m(h_i)=y_i, i=0,1,\cdots,m$，对于 $m=0,1,\cdots,n$．我们希望对于 $m=n+1$ 构造类似的有理函数．因为

$$\frac{P_m(h_i)}{Q_m(h_i)}=y_i, \qquad i\leqslant m\leqslant n, \tag{6.9}$$

试验函数 $T_m(h,y)=P_m(h)-yQ_m(h)$ 在点 (h_i,y_i) 有零

值，$i \leqslant m \leqslant n$。考虑函数

$$T_{n+1}(h, y) = \alpha T_n(h, y) + \beta(h) T_{n-1}(h, y). \quad (6.10)$$

它在点 (h_i, y_i) 一定有零值，$i \leqslant n-1$。如果我们取 $\beta(h_n) = 0$，那么 $T_{n+1}(h_n, y_n) = 0$。现在希望使 $T_{n+1}(h_{n+1}, y_{n+1}) = 0$。这要求

$$\alpha T_n(h_{n+1}, y_{n+1}) + \beta(h_{n+1}) T_{n-1}(h_{n+1}, y_{n+1}) = 0. \quad (6.11)$$

注意到 T_n 和 T_{n-1} 是 $\left[\frac{n+1}{2}\right]$ 和 $\left[\frac{n}{2}\right]$ 次的 h 的多项式，且关于 y 是线性的。我们希望 T_{n+1} 是 $[(n+2)/2]$ 次 h 的多项式且对 y 是线性的，使得我们能够把它表示为 $P_{n+1}(h) + yQ_{n+1}(h)$，其中 $P_{n+1}(h)$ 有 $[(n+1)/2]$ 次且 $Q_{n+1}(h)$ 有 $[(n+2)/2]$ 次。如果在定义关系式(6.10)中 α 是常数且 β 仅是 h 的线性函数，则它将是真确的。于是，我们可以选取 $\beta(h) = (h - h_n)$ 且 α 满足(6.11)：

$$\alpha = (h_n - h_{n+1}) \frac{T_{n-1}(h_{n+1}, y_{n+1})}{T_n(h_{n+1}, y_{n+1})}.$$

代入(6.10)，我们得到

$$\begin{aligned}
P_{n+1}(h) + yQ_{n+1}(h) &= T_{n+1}(h, y) \\
&= (h_n - h_{n+1}) \frac{T_{n-1}(h_{n+1}, y_{n+1})}{T_n(h_{n+1}, y_{n+1})} \\
&\quad \times T_n(h, y) + (h - h_n) T_{n-1}(h, y) \\
&= (h_n - h_{n+1}) \frac{T_{n-1}(h_{n+1}, y_{n+1})}{T_n(h_{n+1}, y_{n+1})} \\
&\quad \times P_n(h) + (h - h_n) P_{n-1}(h) \\
&\quad + y\left[(h_n - h_{n+1}) \frac{T_{n-1}(h_{n+1}, y_{n+1})}{T_n(h_{n+1}, y_{n+1})}\right. \\
&\quad \left. \times Q_n(h) + (h - h_n) Q_{n-1}(h)\right]. \quad (6.12)
\end{aligned}$$

通过使与 y 无关的和与 y 的线性项各自相等的办法，利用 P_n，

$Q_n, P_{n-1}, Q_{n-1}, y_{n+1}, h_{n+1}$和 h_n,给出了 P_{n+1} 和 Q_{n+1} 的递推关系式. 直接使用这个关系式是不方便的，因为它要求每一步计算关于 h 的两个多项式.我们注意到在多项式插值的情况下，公式 (6.4) 给出一个对于 $R^i_m(0) = R^i_m$ 的新值的关系式. 我们希望得到关于 $P_n(0)/Q_n(0)$ 的类似关系式. 已经证明 [Stoer (1961) 和 Bulirsch 和 Stoer (1964)]: 对于有理插值可以导出一种格式.我们定义 $R^i_m(h)$ 在 $h = h_i, h_{i+1}, \cdots, h_{i+m}$ 同 $y(x, h)$ 相同的有理近似, 其中 $h_i > h_{i+1} > \cdots > h_{i+m}$ 且 $R^i_m(0) = R^i_m$, 那么可由下面的公式获得 R:

$$
\begin{aligned}
&R^i_{-1} = 0 \text{ (递推的开始值)},\\
&R^i_0 = y(t, h_i), \qquad\qquad (6.13)\\
&R^i_m = R^{i+1}_{m-1} + \frac{R^{i+1}_{m-1} - R^i_{m-1}}{\left(\frac{h_i}{h_{i+m}}\right)^2\left[1 - \frac{R^{i+1}_{m-1} - R^i_{m-1}}{R^{i+1}_{m-1} - R^{i+1}_{m-2}}\right] - 1},\\
&\qquad m \geqslant 1
\end{aligned}
$$

(这些公式假设,近似值是 h^2 的函数,不是 h 的函数). 这个公式包含计算越来越接近解量的 R^0_m 的差,所以, 通过直接计算差得到小的浮点误差. 定义

$$D^i_m = R^i_m - R^{i+1}_{m-1},$$

$$C^i_m = R^i_m - R^i_{m-1}$$

和

$$W^i_m = R^i_m - R^{i-1}_m,$$

我们可以把 (6.13) 转换成为

$$D^i_m = \frac{C^{i+1}_{m-1} \cdot W^{i+1}_{m-1}}{\left(\frac{h_i}{h_{i+m}}\right)^2 D^i_{m-1} - C^{i+1}_{m-1}}, \qquad m \geqslant 1;$$

$$C^i_m = \frac{\left(\frac{h_i}{h_{i+m}}\right)^2 D^i_{m-1} \cdot W^{i+1}_{m-1}}{\left(\frac{h_i}{h_{i+m}}\right)^2 D^i_{m-1} - C^{i+1}_{m-1}}, \qquad m \geqslant 1;$$

和

$$W_m^i = C_m^i - D_m^{i-1},$$

而

$$C_0^i = D_0^i = y(t, h_i),$$
$$W_0^i = y(t, h_i) - y(t, h_{i-1}).$$

在计算中间不必贮存比包含 $D_{m-i}^i (i=m, m-1, \cdots, 0)$ 的单个数组还多的数组. 这点在下面的 360/FORTRAN 程序中被说明了，这程序利用有理函数或者多项式外插以步长 H 进行积分一步. 此程序是根据 Bulirsch 和 Stoer 的 ALGOL 程序(1966)由Clark (1966) 作的 FORTRAN 译本得来的.

FORTRAN 程序

```
      SUBROUTINE DIFSUB (N,T,Y,DY,H,HMIN,EPS,MF,YMAX,ERROR,KFLAG,
     1                   JSTART,MAXORD,MAXPTS)
      IMPLICIT REAL*8 (A-H,O-Z)
C*********************************************************************
C*   THE PARAMETERS TO THIS INTEGRATION SUBROUTINE HAVE              *
C*   THE FOLLOWING MEANINGS..                                        *
C*     N       THE NUMBER OF FIRST ORDER DIFFERENTIAL EQUATIONS      *
C*     T       THE INDEPENDENT VARIABLE                              *
C*     Y       THE DEPENDENT VARIABLES, UP TO 10 ARE ALLOWED.        *
C*     DY      AN ARRAY OF 10 LOCATIONS WHICH WILL CONTAIN THE       *
C*               VALUES OF THE DERIVATIVES ON EXIT.                  *
C*     H       THE STEP SIZE THAT SHOULD BE ATTEMPTED.  IT MAY BE    *
C*               INCREASED OR DECREASED BY THE SUBROUTINE.           *
C*     HMIN    THE MINIMUM STEP SIZE THAT SHOULD BE ALLOWED ON THIS  *
C*               STEP.                                               *
C*     EPS     THE ERROR TEST CONSTANT.  THE ESTIMATED ERRORS ARE    *
C*               REQUIRED TO BE LESS THAN EPS*YMAX IN EACH COMPONENT. *
C*               IF YMAX IS ORIGINALLY SET TO +1 IN EACH COMPONENT,  *
C*               THE ERROR TEST WILL BE RELATIVE FOR THOSE COMPONENTS *
C*               GREATER THAN 1 AND ABSOLUTE FOR THE OTHERS.         *
C*     MF      THE METHOD INDICATOR.  THE FOLLOWING ARE ALLOWED..    *
C*                     0    BULIRSCH-STOER RATIONAL EXTRAPOLATION    *
C*                     1    POLYNOMIAL EXTRAPOLATION                 *
C*     YMAX    THE MAXIMUM VALUES OF THE DEPENDENT VARIABLES ARE     *
C*               SAVED IN THIS ARRAY. IT SHOULD BE SET TO +1 BEFORE  *
C*               THE FIRST ENTRY. (SEE THE DESCRIPTION OF EPS.)      *
C*     ERROR   THE ESTIMATED SINGLE STEP ERROR IN EACH COMPONENT     *
C*     KFLAG   A COMPLETION CODE WITH THE FOLLOWING MEANINGS..       *
C*                    -1    THE STEP WAS TAKEN WITH H = HMIN         *
C*                          BUT THE REQUESTED ERROR WAS NOT ACHIEVED. *
C*                    +1    THE STEP WAS SUCCESSFUL.                 *
C*     JSTART AN INITIALIZATION INDICATOR WITH THE MEANING..         *
C*                    -1    REPEAT THE LAST STEP, RESTORING THE      *
C*                          VALUES OF Y AND YMAX THAT WERE USED      *
C*                          LAST TIME.                               *
C*                    +1    TAKE A NEW STEP.                         *
C*     MAXORD THE MAXIMUM ORDER OF EXTRAPOLATION ALLOWED. IT MUST    *
C*               BE LESS THAN 11.                                    *
C*     MAXPTS THE MAXIMUM NUMBER OF DIFFERENT SUB STEP SIZES USED    *
C*               IN THE EXTRAPOLATION PROCESS.                       *
C*********************************************************************
```

```
      DIMENSION Y(10),DY(10),YMAX(10),YSAVE(10),YNM1(10),YN(10),DYN(10)
     1          ,YMAXSV(10),QUOT(11,2),EXTRAP(10,11),YNM1HV(10,12)
     2          ,YNHV(10,12),YMAXHV(10,12),ERROR(10)
C*********************************************************************
C*   THE ARRAYS ARE USED FOR THE FOLLOWING DATA..                    *
C*     YSAVE   THE INITIAL VALUES OF Y ARE SAVED FOR A RESTART       *
C*     YNM1    Y(N-1), THE PREVIOUS VALUE OF Y IN THE MIDPOINT METHOD  *
C*     YN      Y(N), THE CURRENT VALUE OF Y IN THE MIDPOINT INTEGRATION *
C*     DYN     THE INITIAL VALUE OF THE DERIVATIVE OF Y.             *
C*     YMAXSV  THE SAVED VALUES OF YMAX AT THE INITIAL POINT.        *
C*     QUOT    THE QUOTIENTS (H(I)/H(I+M))**2 USED IN THE EXTRAPOLATION.*
C*     EXTRAP  THE MOST RECENT EXTRAPOLATED VALUES OF Y IN THE CASE  *
C*              OF POLYNOMIAL EXTRAPOLATION, OR OF THE DIFFERENCES IN *
C*              THE CASE OF RATIONAL FUNCTION EXTRAPOLATION.         *
C*     YNM1HV  THE VALUES OF YNM1 AT THE MIDPOINT OF THE BASIC INTERVAL *
C*              IF THE NUMBER OF SUB STEPS IS DIVISIBLE BY 4. THIS    *
C*              INFORMATION IS USED TO AVOID REDOING THE INTEGRATION IN *
C*              CASE THE STEP IS HALVED.                             *
C*     YNHV    THE SIMILAR VALUES OF YN                              *
C*     YMAXHV  AND THE SAME FOR YMAX                                 *
C*     ERROR   THE ESTIMATES OF THE SINGLE STEP ERRORS ARE SAVED HERE. *
C*********************************************************************
      DATA QUOT/1.,2.25,4.,9.,16.,36.,64.,144.,256.,576.,1024.,
     1     1.,1.7777777777777777,4.,7.1111111111111111,
     2     16.,28.4444444444444,64.,113.7777777777777,
     3      256.,455.1111111111111,1024./
      DATA FMAX/10000000./
C*********************************************************************
C*   FMAX IS A NUMBER SMALLER THAN THE FIRST INTEGER THAT CANNOT BE  *
C*   REPRESENTED EXACTLY IN FLOATING POINT.                          *
C*********************************************************************
      IF(JSTART.LT.0) GO TO 2
C*********************************************************************
C* SAVE THE VALUES OF Y AND YMAX IN CASE A RESTART IS NECESSARY.     *
C*********************************************************************
      DO 1 I = 1,N
      YSAVE(I) = Y(I)
    1 YMAXSV(I) = YMAX(I)
      CALL DIFFUN(T,Y,DYN)
      GO TO 4
C*********************************************************************
C*   RESTORE THE VALUES OF Y AND YMAX FOR A RESTART.                 *
C*********************************************************************
    2 DO 3 I = 1,N
      Y(I) = YSAVE(I)
    3 YMAX(I) = YMAXSV(I
    4 CONTINUE
C*********************************************************************
C*   THE FOLLOWING COUNTERS AND SWITCHES ARE USED..                  *
C*     J       IS THE COUNT THROUGH THE DIFFERENT SUB STEPS G USED.  *
C*     JODD    IS 1 IF J IS ODD, 2 IF J IS EVEN                      *
C*     JHVSV   IS THE NUMBER OF SUBSTEP SIZES FOR WHICH HALF WAY     *
C*              INFORMATION HAS BEEN SAVED.                          *
C*     JHVSV1  THE VALUE OF JHVSV FROM THE PREVIOUS CYCLE            *
C*     M       THE NUMBER OF PAIRS OF SUB STEPS WHICH MAKE UP THE STEP H*
C*              M TAKES THE SEQUENCE 1,2,3,4,6,8,12,16, ETC.         *
C*     MNEXT   THE NEXT VALUE OF M                                   *
C*     MTWO    THE NEXT BUT ONE VALUE OF M.                          *
C*     QUOTSV  THE LAST VALUE OF QUOT IS IRREGULAR DUE TO THE FACT THAT *
C*              THE SEQUENCE BY THE MULTIPLES 9/4,16/9 (ODD) OR      *
C*              16/9,9/4 (EVEN UNTIL THE FINAL MULTIPLE OF 4. HOWEVER, *
C*              (H(0)/H(M))**2 IS ALWAYS M**2. THE REGULAR VALUE OF  *
C*              QUOT IS SAVED IN QUOTSV, AND REPLACED BY M**2.       *
C*     KONV    IS SET TO +1 INITIALLY, AND RESET TO -1 IF THE ERROR  *
C*              TEST FAILS.                                          *
C*********************************************************************
    5 JHVSV1 = 0
      KFLAG = 1
    6 JHVSV = 0
    7 A = H + T
      JODD = 1
      M = 1
      MNEXT = 2
      MTWO = 3
      DO 23 J = 1,MAXPTS
        QUOTSV = QUOT(J,JODD)
        QUOT(J,JODD) = M*M
        KONV = 1
        IF (J.LE.(MAXORD/2)) KONV = -1
```

```
      IF (J.LE.(MAXORD+1))  GO TO 8
      L = MAXOR( + 1
      HCHNGE = .7071068D0*HCHNGE
      GO TO 9
 8    L = J
      HCHNGE = 1.0D0 + (MAXORD + 1 - J)/6.0D0
 9    B = H/M
      G = B*0.5D0
      IF (J.GT.JHVSV) GO TO 11
C*********************************************************************
C*   THE VALUES OF THE MIDPOINT INTEGRATION WERE SAVED AT THE       *
C*   HALF WAY POINT IN THE PREVIOUS INTEGRATION. USE THEM.          *
C*********************************************************************
      DO 10 I = 1,N
        YN(I) = YNHV(I,J)
        YNM1(I) = YNM1HV(I,J)
 10     YMAX(I) = YMAXHV(I,J)
      GO TO 16
C*********************************************************************
C*   INTEGRATE OVER THE RANGE H BY 2*M STEPS OF A MIDPOINT METHOD.  *
C*********************************************************************
 11   DO 12 I = 1,N
        YNM1(I) = YSAVE(I)
        YN(I) = YSAVE(I) + G*DYN(I)
 12     YMAX(I) = YMAXSV(I)
      M2 = M + M
      TU = T
      DO 15 K = 2,M2
        TU = TU + G
        CALL DIFFUN(TU,YN,DY)
        DO 13 I = 1,N
          U = YNM1(I) + B*DY(I)
          YNM1(I) = YN(I)
          YN(I) = U
          U = DABS(U)
 13       IF (U.GT.YMAX(I)) YMAX(I) = U
        IF ((K.NE.M).OR.(JHVSV.NE.0).OR.(K.EQ.3)) GO TO 15
        JHVSV = JHVSV + 1
        DO 14 I = 1,N
          YNHV(I,JHVSV) = YN(I)
          YNM1HV(I,JHVSV) = YNM1(I)
 14       YMAXHV(I,JHVSV) = YMAX(I)
 15     CONTINUE
 16   CALL DIFFUN(A,YN,DY)
      DO 22 I = 1,N
        V = EXTRAP(I,1)
C*********************************************************************
C*   CALCULATE THE FINAL VALUE TO BE USED IN THE EXTRAPOLATION PROCESS *
C*********************************************************************
        TA = (YN(I) + YNM1(I) + G*DY(I))*0.5D0
        C = TA
C*********************************************************************
C*   INSERT THE INTEGRAL AS THE FIRST EXTRAPOLATED VALUE.           *
C*********************************************************************
        EXTRAP(I,1) = TA
        IF (L.LT.2) GO TO 21
        IF (DABS(V)*FMAX.LT.DABS(C)) GO TO 27
        IF (MF.GT.0) GO TO 19
C*********************************************************************
C*   PERFORM THE EXTRAPOLATION BY RATIONAL FUNCTIONS ON THE         *
C*   SECOND AND SUBSEQUENT INTEGRALS.                               *
C*********************************************************************
        DO 18 K = 2,L
          B1 = QUOT(K,JODD)*V
          B = B1 - C
          U = V
          IF (B.EQ.0) GO TO 17
          B = (C - V)/B
          U = C*B
          C = B1*B
 17       V = EXTRAP(I,K)
          EXTRAP(I,K) = U
          TA = TA + U
 18       CONTINUE
        GO TO 21
C*********************************************************************
C*   PERFORM THE EXTRAPOLATION BY POLYNOMIALS ON THE                *
C*   SECOND AND SUBSEQUENT INTEGRALS.                               *
C*********************************************************************
```

应用这程序用 $H=1$ 从 0 到 1 积分 $y'=-y$，关于 R^{l}_{m} 的误差乘以 10^9，在表 6.7 中表示. 它们比表 6.5 中的值稍大些，虽然它们右函数计算次数相同且在外插过程中稍微多一些运算. 但是，对于其他问题就不是这样，如图 6.1 中说明的那样. 此例取自 Clark (1966). 它通过各种要求的误差对函数数求值次数的图象说明在三个问题中的误差. 在每一种情况 MAXORD 给出为 6，MAXPTS 给出为 10. 负指数问题是 $y'=-y$，$y(0)=1$，积分到 $t=20$，在每积分一步以前用 YMAX 的值顶掉 y 的当前值. 相对于 $\exp(-20)$ 的误差曲线划出来了.

表 6.7 Bulirsch-Stoer 方法的误差

l	h_l m	0	1	2	3	4	5
0	$\frac{1}{2}$	7 120 559	1 912 226	60 374	953	7	0
1	$\frac{1}{4}$	3 214 309	266 135	4 667	33	0	
2	$\frac{1}{6}$	1 576 434	70 034	548	2		
3	$\frac{1}{8}$	817 384	17 919	79			
4	$\frac{1}{12}$	417 681	4 539				
5	$\frac{1}{16}$	236 932					

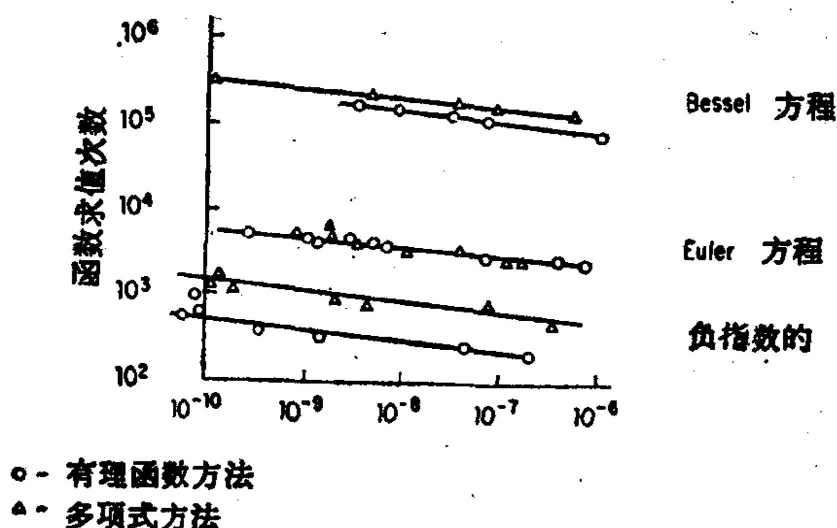
Euler 方程是如下方程组:

$$y^{1\prime}=-y^2y^3,\qquad y^1(0)=0,$$
$$y^{2\prime}=-y^1y^3,\qquad y^2(0)=1,$$
$$y^{3\prime}=-0.51\,y^1y^2,\qquad y^3(0)=1,$$

积分到 $t=60$. 划出三个 y 的均方根误差曲线. Bessel 方程利用

$$y^{1\prime}=y^2,$$
$$y^{2\prime}=\left(\frac{256}{t^2}-1\right)y^1-\frac{y^2}{t},$$
$$y^1(6)=1.20195\cdot 10^{-6},$$
$$y^2(6)=2.98648\cdot 10^{-6}$$

定义 $J_{16}(t) = y^1(t)$，大约在 $t = 18.1$ 误差被取成相对于 y^1 的最大值，在那儿 $y^1 \cong 0.2612$. 在 $t = 6132, 6134, 6136$ 和 6138，划出平均绝对误差曲线.



○ - 有理函数方法

△ - 多项式方法

图 6.1. 有理函数与多项式外插的比较

表 6.8. y_i 对 EPS 参数的误差 e

EPS	e_1	e_2	e_3	e_4	e_5
0.001	3235353	5135388	2981195	0	1450333913
0.0001	7724	2891	2832	0	32023507
0.00001	17	15	4	0	18740
0.000001	17	15	4	0	18740
0.0000001	0	0	0	0	6
0.00000001	0	0	0	0	6
0.000000001	0	0	0	0	6
0.0000000001	0	0	0	0	1

我们可以看到，其中两个例子有理函数插值好一些.

对于有理函数外插，减少误差控制参数 EPS 的影响表示在表 6.8 中. 下面的方程组表示，所要求的误差对实际误差乘 10^9：

$$y^{1\prime} = -y^1, \qquad y^1(0) = 1$$
$$y^{2\prime} = y^3, \qquad y^2(0) = 0$$
$$y^{3\prime} = -y^2, \qquad y^3(0) = 1$$

$$y^{4\prime} = 2t, \qquad y^4(0) = 0$$
$$y^{5\prime} = 10t^9, \qquad y^5(0) = 0$$

总之,外插方法看起来是有希望的. 如果比较图 6.1 和图 5.2, 可以看出比 Runge-Kutta 方法优越. 但是,关系到稳定性、误差估计和误差界,有许多未解决的问题.

问　　题

1. 在区间 [0, 1] 上积分 $y' = y$, $y(0) = 1$, 利用如下方法:
   (a) 中点法则(等式 2.5);
   (b) Heun 法则(等式 2.8);
   (c) 梯形法则(等式 2.7).
   在这些方法的基础上, 进行多项式外插, 所用步长 $h = 1, 1/2, 1/3, 1/4, 1/6, 1/8, 1/12$, 并依据假设: 解是缺线性项的关于 h 的幂级数;解是 h^2 的幂级数. 你们的结论是什么?
2. 如果使用由算式(6.4)所给出的那种多项式外插,列出一个类似于表 6.1 的表,如果比值 h_i/h_{i+1} 是 2 且假设 $y(t, h_i)$ 的舍入误差 $\leqslant 2^i e$, 说明舍入误差的最坏情况.
3. 对方程 $y' = -2^{10}y$, $y(0) = 1$, 在区间 [0,1] 上使用多项式外插且试验性地决定方法是绝对稳定时 H 的近似值, 如果取最后的解为 R_m^i, $i + m \leqslant 5$, 其中 R_m^i 由(6.4)给出,对于基本的积分使用 Euler 方法且所用的步长是 2^iH.
4. 在区间 H 用步长 H, $H/2$ 和 $H/4$, 用Euler 方法去确定 R_0^0, R_0^1 和 R_0^2 作为微分方程 $y' = \lambda y$ 的解的近似值, 现在用等式(6.4)计算 R_2^0. 导出 $H\lambda$-平面上确定绝对稳定区域的方程.

7. 多值或多步方法——导论

至此所讨论的方法仅仅要求微分方程的知识和初始值. 因此,给出一个对 $y(t)$ 的值在 $t=t_{n-1}$ 的近似值,比如说 y_{n-1},它们已经提供了计算 $y_n \cong y(t_n)$ 的一种方法. 于是它们可以被称之为单值方法, 因为它们仅仅要求因变量的一个值. 但它们通常称为单步方法, 因为它们只要求一个节点的值去计算下一个值. 一旦已计算了数值近似的许多点的值, 可用它们来帮助计算后面点的值 (例如用多项式外插), 很象在 Runge-Kutta 方法中用的中间点值一样. 来自前面点,比如 $t \leqslant t_{n-1}$ 的信息可以贮存并表示关于解在 $t=t_{n-1}$ 的知识.在下面四章里我们将要讨论在 $t=t_{n-1}$ 要求几部分关于因变量的信息的方法, 以便计算在 $t=t_n$ 相应的部分的信息. 因此, 我们称这些方法为多值方法, 因为它们使用超过一个因变量的值. 这些方法常常利用因变量和它的导数在 k 个不同节点 $t_{n-1}, t_{n-2}, \cdots, t_{n-k}$ 的值,所以它们通常被称为多步方法, 在这种情况下,亦称为 k 步方法.

在这一章里我们将提出一类多值方法的记法. 然后研究多值方法的两个特殊情况,即显式和隐式的多步方法. 特别, 我们将研究 Adams-Bashforth 和 Adàms-Moulton 方法,它们是这种情况的例子. 我们将仅仅讨论单个方程 $y'=f(y)$. 显然全部情形可推广到方程组.

7.1. 多 值 方 法

在一些点,比如说 $t_{n-k}, t_{n-k+1}, \cdots$ 和 t_{n-1} 点计算了近似

值之后，得到值 $y_{n-k}, y_{n-k+1}, \cdots, y_{n-1}, hy'_{n-k}, hy'_{n-k+1}, \cdots$ 和 hy'_{n-1}. 我们利用这些信息，或者它们的子集帮助确定 y_n 和 $hy'_n = hf(y_n)$. 我们记 $\mathbf{y}_{n-1}$ 为列向量

$$[y_{n-1}, y_{n-2}, \cdots, y_{n-k}, hy'_{n-1}, hy'_{n-2}, \cdots, hy'_{n-k}]^T$$

(计算 y_n 时不用的分量抹去). 一个多值方法的目标是从 $\mathbf{y}_{n-1}$ 和微分方程求得对于 $\mathbf{y}_n$ 的数值近似，其中 $\mathbf{y}_n$ 是列向量 $[y_n, y_{n-1}, \cdots, y_{n-k+1}, hy'_n, hy'_{n-1}, \cdots, hy'_{n-k+1}]^T$. 一旦给出 $\mathbf{y}_0$, 可以重复应用这个过程计算 $\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N$.

计算 $\mathbf{y}_0$ 的问题，常常使人们避免使用多值方法而赞成使用单值方法，但是对于要求有好的精度的大的问题，多值方法比单值方法所提高的速度是显著的. 因为我们仅仅给了 y_0 作为初值，还必须计算我们称之为开始值的$\mathbf{y}_0$的其他分量. 通常的办法是利用一单值方法，比如 Runge-Kutta 方法来计算 $y_1, y_2, \cdots, y_{k-1}$, 然后计算 $hy'_i = hf(y_i)$, $0 \leqslant i < k$. 因此，构成 $\mathbf{y}_{k-1}$ 作为开始值以便计算 $\mathbf{y}_k, \mathbf{y}_{k+1}, \cdots$. 我们将假设已经使用了某种这样的办法，虽然在第 9 章提出一个程序，它避免了多值方法的"开始值问题".

一多值方法包含我们称之为预估和校正的两个过程. 在预估过程中，由 $\mathbf{y}_{n-1}$ 用线性外插计算对 $\mathbf{y}_n$ 的近似值. 我们称这个近似为 $\mathbf{y}_{n,(0)}$ 且由

$$\mathbf{y}_{n,(0)} = B\mathbf{y}_{n-1} \tag{7.1}$$

给出，B 是任意一适当的常数矩阵. 例如它可以是 124 页所示的其中 $\alpha_i, \beta_i, \gamma_i$ 和 δ_i 是常数，使得

$$\sum_{i=1}^{k} (\alpha_i y_{n-i} + \beta_i hy'_{n-i})$$

是对 y_n 的近似且

$$\sum_{i=1}^{k} (\gamma_i y_{n-i} + \delta_i hy'_{n-i})$$

是对 hy'_n 的近似.

$$
\begin{bmatrix} y_{n,(0)} \\ y_{n-1} \\ \vdots \\ y_{n-k+1} \\ hy'_{n,(0)} \\ \vdots \\ hy'_{n-k+1} \end{bmatrix}
=
\left[\begin{array}{cccc|cccc}
\alpha_1 & \alpha_2 & \cdots & \alpha_k & \beta_1 & \beta_2 & \cdots & \beta_k \\
1 & & 0 & & & & & \\
 & \ddots & & & & 0 & & \\
0 & \cdots & 1 & 0 & & & & \\
\hline
\gamma_1 & \gamma_2 & & \gamma_k & \delta_1 & \delta_2 & \cdots & \delta_k \\
 & & & & 1 & & 0 & \\
 & 0 & & & & \ddots & & \\
 & & & & 0 & & 1 & 0
\end{array}\right]
\begin{bmatrix} y_{n-1} \\ y_{n-2} \\ \vdots \\ y_{n-k} \\ hy'_{n-1} \\ \vdots \\ hy'_{n-k} \end{bmatrix}
$$

预估过程不以任何方式使用微分方程,所以,如果在 $t=t_n$ 它们不满足微分方程,可以用校正过程校正近似值.微分方程写成为

$$0=G(\mathbf{y}_n)=-(\mathbf{y}_n)_k+hf((y_n)_0)=-hy'_n+hf(y_n),$$

这里我们用 $(\mathbf{y})_i$ 表示向量 $\mathbf{y}$ 的第 i 个分量(向量从 0 开始标号). $\mathbf{y}_{n,(0)}$ 不满足微分方程的量是 $G(\mathbf{y}_{n,(0)})$. 一向量乘以这个纯量加到 $\mathbf{y}_{n,(0)}$, 按照下面过程:

$$\mathbf{y}_{n,(1)}=\mathbf{y}_{n,(0)}+\mathbf{c}G(\mathbf{y}_{n,(0)}) \tag{7.2}$$

来校正它. 对固定的迭代次数或者在 $\mathbf{y}_{n,(m)}$ 没有进一步改变以前可以用

$$\mathbf{y}_{n,(m+1)}=\mathbf{y}_{n,(m)}+\mathbf{c}G(\mathbf{y}_{n,(m)}),\ m=1,2,\cdots \tag{7.3}$$

重复这个过程. 然后对 $\mathbf{y}_n$ 所使用的值是 $\mathbf{y}_{n,(M)}$, 其中 M 或者是固定的或者是达到收敛的足够大的值, 即使得对所要求的精度而言, $G(\mathbf{y}_{n,(M)})$ 为零. 我们说 M 是校正迭代的次数.

7.2. 显式多步方法——Adams-Bashforth 方法

假如 B 有上面给出的形式,具有 $\gamma_i=\delta_i=0,\ 1\leqslant i\leqslant k$, 且向量 $\mathbf{c}$ 的所有位置上,除了第 k 个置 1 外,全为零. 在这种

情形,(7.1)是

$$y_{n,(0)}=\sum_{i=1}^{k}(\alpha_i y_{n-i}+\beta_i h y'_{n-i}),$$
$$hy'_{n,(0)}=0,$$

同时(7.2)给出

$$y_{n,(1)}=y_{n,(0)},$$
$$hy'_{n,(1)}=hf(y_{n,(0)}).$$

附加的迭代(7.3)没有更多的效果,所以,我们可以取 $M=1$ 且得到.

$$y_n=\sum_{i=1}^{k}(\alpha_i y_{n-i}+\beta_i h y'_{n-i}), \tag{7.4}$$
$$hy'_n=hf(y_n).$$

这称为显式的多步方法,因为它提供了一个根据前面许多点的 y 和它的导数值计算 y_n 和 hy'_n 的显式方法.每一步它仅仅要求一个向量内积和对 f 的一次求值.

Adams-Bashforth 方法[Bashforth 和 Adams(1883)]是这种方法的特例.

我们按三种不同的办法来推导它,为后面更加一般的方法的讨论打下基础.最简单的推导是通过如下积分的办法.积分

$$y'=f(y,t)=f(t)$$

得到

$$\int_{t_{n-1}}^{t_n} y'dt=\int_{t_{n-1}}^{t_n} f(t)dt$$

或者

$$y(t_n)=y(t_{n-1})+\int_{t_{n-1}}^{t_n} f(t)dt, \tag{7.5}$$

我们可以通过一些在 $t=t_{n-1}, t_{n-2}, \cdots$ 的已知值,比如说 $f_{n-1}, f_{n-2}, \cdots$,用一插值多项式来近似它.我们将使用 Newton

向后差分公式[见 Hildebrand (1956), §4.3 节]. 如果 $f(t)$ 有连续的 k 阶导数，$t_m = t_0 + mh$，$f_m = f(t_m)$，且向后差分用

$$\nabla^{q+1} f_m = \nabla^q f_m - \nabla^q f_{m-1}$$

给出，其中 $\nabla^0 f_m = f_m$，那么

$$f(t) = f_m + \frac{(t-t_m)\nabla f_m}{h} + (t-t_m)(t-t_{m-1})\frac{\nabla^2 f_m}{2!h^2}$$

$$+ \cdots + (t-t_m)\cdots(t-t_{m-k+2})\frac{\nabla^{k-1} f_m}{(k-1)!h^{k-1}}$$

$$+ (t-t_m)\cdots(t-t_{m-k+1})\frac{f^{(k)}(\xi)}{k!}, \qquad (7.6)$$

其中 $f^{(k)}(\xi)$ 是 f 的 k 阶导数，在包含 t，t_{m-k+1} 和 t_m 的区间内的某点求值.如果令 $s = (t - t_{n-1})/h$ 且 $m = n - 1$，(7.6) 变成

$$f(t) = \binom{-s}{0} f_{n-1} - \binom{-s}{1} \nabla f_{n-1} + \cdots$$

$$+(-1)^{k-1}\binom{-s}{k-1}\nabla^{k-1} f_{n-1} + (-1)^k h^k \binom{-s}{k} f^{(k)}(\xi),$$

其中

$$\binom{s}{q} = \frac{s(s-1)\cdots(s-q+1)}{q!} \text{和} \binom{s}{0} = 0.$$

把它代入到 (7.5)，我们得到

$$y(t_n) = y(t_{n-1}) + \int_{t_{n-1}}^{t_n} \left[\sum_{j=0}^{k-1} (-1)^j \binom{-s}{j} \nabla^j f_{n-1}\right.$$

$$\left. + (-1)^k h^k \binom{-s}{k} y^{(k+1)}(\xi)\right] dt$$

或者

$$y(t_n) = y(t_{n-1}) + h\sum_{j=0}^{k-1} \gamma_j \nabla^j f_{n-1} + (-1)^k h^{k+1}$$

$$\times \int_0^1 \binom{-s}{k} y^{(k+1)}(\xi) ds, \tag{7.7}$$

其中

$$\gamma_j = (-1)^j \int_0^1 \binom{-s}{j} ds. \tag{7.8}$$

如果忽略(7.7)中最后一项,我们得到 k 步 Adams-Bashforth 公式:

$$y_n = y_{n-1} + h \sum_{j=1}^{k-1} \gamma_j \nabla^j f_{n-1}. \tag{7.9}$$

它利用在 t_{n-1} 的值和在 t_{n-1} 的导数的向后差分表示 $y(t)$ 在 t_n 的值的近似. 利用前面许多点的值可以表示向后差分, 利用公式:

$$\nabla^q f_{n-1} = \sum_{i=1}^{q} (-1)^i \binom{q}{i} f_{n-1-i}.$$

这样,(7.9) 可以再表示为

$$y_n = y_{n-1} + h \sum_{i=1}^{k} \beta_{ki} f_{n-i}, \tag{7.10}$$

其中

$$\beta_{ki} = (-1)^{i-1} \sum_{j=i-1}^{k-1} \gamma_j \binom{j}{i-1}. \tag{7.11}$$

方程(7.10)利用在 $t_{n-1}, t_{n-2}, \cdots, t_{n-k}$ 的信息表示 y_n,所以,它被看作是 k 步方法,虽然我们也可以称它为 $k+1$ 值方法,因为在计算中利用了关于解的性质的 $k+1$ 项信息. 当我们令 $\alpha_1 = 1, \alpha_2 = \alpha_3 = \cdots = \alpha_k = 0$ 和 $\beta_i = \beta_{ki}$ 时, 它就相当于(7.4). 通过构造一个 f 的差分表, 公式(7.9)可以用来代替(7.10).两方法是等价的,不同点仅在算术运算的次数和舍入误差方面. 这些论题将在第 9 章讨论.

以后我们正式定义局部截断误差是真解和从精确值开始一步所提供的解之间的差. 象在单步方法中一样, 阶 r 应使

得局部截断误差为 $O(h^{r+1})$ 的整数。我们将看到 k 步 Adams-Bashforth 方法的阶是 k，因为从 (7.7) 到 (7.9) 中被忽略的项是 $O(h^{r+1})$. 但是注意，不管 k 如何，每步仅要求 f 的一次求值。这大大不同于象 R-k 那样的高阶单步方法。

例子

从 (7.8) 我们看到

$$\gamma_0 = 1,$$
$$\gamma_1 = \frac{1}{2},$$

因而二阶公式是

$$y_n = y_{n-1} + h\left(f_{n-1} + \frac{1}{2}\nabla f_{n-1}\right)$$

或者

$$y_n = y_{n-1} + h\left(\frac{3}{2}f_{n-1} - \frac{1}{2}f_{n-2}\right). \tag{7.12}$$

用下面的 FORTRAN 程序从 $y(0)=1$ 到 $t=1$ 积分方程 $y'=-y$，并用 $h=2^{-k}$, $k=1, 2, \cdots, 7$. 结果表示在表 7.1 中。

表 7.1 用二阶 Adams-Bashforth 方法积分 $y'=-y$

H	Y	误差	误差/H**2
0.50000E 00	0.40163E 00	−0.33753E−01	−0.13501E 00
0.25000E 00	0.37628E 00	−0.83983E−02	−0.13437E 00
0.12500E 00	0.37014E 00	−0.22579E−02	−0.14451E 00
0.62500E−01	0.36846E 00	−0.58228E−03	−0.14906E 00
0.31250E−01	0.36803E 00	−0.14663E−03	−0.15015E 00
0.15625E−01	0.36791E 00	−0.34988E−04	−0.14331E 00
0.78125E−02	0.36788E 00	−0.47684E−05	−0.78125E−01

```
WRITE(6,4)
DO 2 K = 1,7
H = 2.0**(-K)
N = 2**K - 1
```

```
      Y = EXP(-H)
      FOLD = -H
      DO1I = 1, N
      F = -H*Y
      Y = Y + 1.5*F - 0.5*FOLD
    1 FOLD = F
      ERROR = EXP(-1.0) - Y
      ERRBYH = ERROR/H**2
    2 WRITE(6, 3)H, Y, ERROR, ERRBYH
      RETURN
    3 FORMAT(4E20.5)
    4 FORMAT('1', 13X, 'H', 19X, 'Y', 17X, 'ERROR', 13X,
       'ERROR/H**2'/)
      END
```

从最后一列可以看出,结果的误差是 $O(h^2)$.(在 IBM 360 上按单倍精度进行这个计算时，最后 h 的突然减少是由于舍入误差碰巧抵消的缘故,这舍入误差大约是 10^{-6}.)

7.2.1. 系数的生成函数

方程(7.8)和(7.11)可以用来确定系数 γ_j 和 β_{kj}，但是它们并没有给出最方便的形式．而生成函数的方法通常是最方便的．用

$$G(t) = \sum_{j=0}^{+\infty} \gamma_j t^j$$

定义函数 $G(t)$.

对 $|t| < 1$ 求和是绝对收敛的,因为根据(7.8) $\gamma_j \leqslant 1$. 因此

$$G(t) = \sum_{j=0}^{\infty} (-1)^j \int_0^1 \binom{-s}{j} ds (t)^j$$

$$= \int_0^1 \sum_{j=0}^{\infty} (-t)^j \binom{-s}{j} ds$$

$$= \int_0^1 (1-t)^{-s} ds$$

$$= -\frac{1}{\log(1-t)}[(1-t)^{-s}]_0^1$$

$$= -\frac{t}{(1-t)\log(1-t)},$$

所以

$$-\frac{\log(1-t)}{t}G(t) = \frac{1}{1-t}$$

或者

$$\left(1+\frac{1}{2}t+\frac{1}{3}t^2+\cdots\right)(\gamma_0+\gamma_1 t+\gamma_2 t^2+\cdots)$$
$$=1+t+t^2+\cdots.$$

使 t^m 的系数相等,我们得到

$$\gamma_m+\frac{1}{2}\gamma_{m-1}+\frac{1}{3}\gamma_{m-2}+\cdots+\frac{1}{m+1}\gamma_0=1.$$

这提供了系数的递推公式. 用它得到下面的 γ_m 的值:

表 7.2 Adams-Bashforth 方法的系数

m	0	1	2	3	4	5
γ_m	1	$\frac{1}{2}$	$\frac{5}{12}$	$\frac{3}{8}$	$\frac{251}{720}$	$\frac{95}{288}$

将这些值用于(7.11),我们得到 β_{ki} 的下面值:

表 7.3 Adams-Bashforth 方法的系数

i	1	2	3	4	5	6
β_{1i}	1					
$2\beta_{2i}$	3	−1				
$12\beta_{3i}$	23	−16	5			
$24\beta_{4i}$	55	−59	37	−9		
$720\beta_{5i}$	1901	−2774	2616	−1274	251	
$1440\beta_{6i}$	4277	−7923	9982	−7298	2877	−475

应该注意，大多数 β_{kj} 超过 1，它们将有放大任一舍入误差的效果.

7.2.2. 推导 Adams-Bashforth 方法的另外两个办法

推导方法的一个办法是待定系数法.假定这方法的形式，比如说是

$$y_n = \alpha_1 y_{n-1} + h\beta_1 f_{n-1} + h\beta_2 f_{n-2}, \tag{7.13}$$

并且选取 α 和 β，使得用 y_{n-1}，f_{n-1} 和 f_{n-2} 的正确值由 (7.13) 所计算的 y_n 值与真解相差尽可能的小. 在这种情形，我们希望误差是 $O(h^3)$，所以，按具有 $O(h^3)$ 余项的 Taylor 级数展开 $y(t_{n-q})$ 和 $y'(t_{n-q})$，就得到

$$\begin{aligned} y(t_n) = \alpha_1 \Big[& y(t_n) - hy'(t_n) + \frac{h^2}{2} y''(t_n) + \frac{h^3}{2} \\ & \times \int_0^{-1} (1+\tau)^2 y^{(3)}(t_n + \tau h) d\tau \Big] + \beta_1 h \\ & \times \Big[y'(t_n) - hy''(t_n) - h^2 \int_0^{-1} (1+\tau) y^{(3)}(t_n + \tau h) d\tau \Big] \\ & + \beta_2 h \Big[y'(t_n) - 2hy''(t_n) \\ & - h^2 \int_0^{-2} (2+\tau) y^{(3)}(t_n + \tau h) d\tau \Big], \end{aligned} \tag{7.14}$$

使 h^0，h^1 和 h^2 的项相等，我们得到

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 1 \\ -\frac{1}{2} & -1 & -2 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \beta_1 \\ \beta_2 \end{bmatrix},$$

有解

$$\alpha_1 = 1, \qquad \beta_1 = \frac{3}{2}, \qquad \beta_2 = -\frac{1}{2}.$$

如所期望的一样，它导出二阶的 Adams-Bashforth 方法．导出这些方程的一个更简单的办法，是要求方法对所有的二次（对这种情形）和小于二次的多项式是精确的．于是，我们可以将 $y=1$，s 和 s^2 代入函数中，其中 $s=(t-t_n)/h$．显然这与要求 Taylor 级数前三项抵消的办法是等价的，且导致同样的方程．

7.2.3. Adams-Bashforth 方法的截断误差

我们为什么要考虑推导这些公式的不同办法？最后的办法是推导这些方程的最简单的方法．这个办法可以用来得到形如

$$y_n=\sum_{i=1}^{k}\alpha_i y_{n-i}+h\sum_{i=1}^{k}\beta_i f_{n-i} \tag{7.15}$$

的一般方法的系数，同时或许是生成 α 和 β 必须满足的线性方程的最简单的办法（后面我们将看到，其他考虑也许会限制在选取 α 和 β 系数方面的自由度）．

第二个方法可能给出方法的局部截断误差的界．它可以应用于类似 (7.15) 的一般方法．如果在 Taylor 级数方法中使用余项定理的积分形式，则可得到余项的一个精确形式．例如从 (7.14) 我们有

$$\begin{aligned}y(t_n)-\alpha_1 y(t_{n-1})-h[\beta_1 y'(t_{n-1})+\beta_2 y'(t_{n-2})]\\ =\frac{h^3}{2}\left[\int_0^{-1}(\alpha_1(1+\tau)^2-2\beta_1(1+\tau))y^{(3)}(t_n+\tau h)d\tau\right.\\ \left.-\int_0^{-2}2\beta_2(2+\tau)y^{(3)}(t_n+\tau h)d\tau\right].\end{aligned}$$

记左边为 $L_n(y(t_n))$，我们有

$$|L_h(y(t_n))|=\frac{h^3}{2}\left|\int_0^{-2}G(\tau)y^{(3)}(t_n+\tau h)d\tau\right|,$$

其中

$$G(\tau)=\begin{cases}\alpha_1(1+\tau)^2-2\beta_1(1+\tau)-2\beta_2(2+\tau) \\ \qquad\qquad\qquad\qquad\qquad 0\geqslant\tau\geqslant-1, \\ -2\beta_2(2+\tau) \qquad\qquad -1>\tau\geqslant-2.\end{cases}$$

因此，局部截断误差以

$$|L_h(y(t_n))|\leqslant-\frac{h^3}{2}\int_0^{-2}|G(\tau)||y^{(3)}(t_n+h\tau)|d\tau$$

$$\leqslant-\frac{h^3}{2}\left[\int_0^{-2}|G(\tau)|d\tau\right]\max_{t_n\geqslant t\geqslant t_{n-2}}|y^{(3)}(t)|$$

或者

$$|L_h(y(t_n))|\leqslant Mh^3|y^{(3)}(\xi)| \tag{7.16}$$

为界，其中

$$M=-\frac{1}{2}\int_0^{-2}|G(\tau)|d\tau.$$

对于 Adams-Bashforth 两步方法，

$$G(\tau)=\tau^2 \qquad 0\geqslant\tau\geqslant-1,$$

$$G(\tau)=2+\tau \qquad -1>\tau\geqslant-2.$$

因此

$$M=-\frac{1}{2}\int_0^{-2}|G(\tau)|d\tau=\frac{5}{6}. \tag{7.17}$$

推导 Adams-Bashforth 方法的第一个方法直接给我们这个界．引用 (7.7) 中使用的 Newton 向后差分的余项形式，我们得到

$$L_h(y(t_n))=(-1)^k h^{k+1}\int_0^1\binom{-s}{k}y^{(k+1)}(\xi)ds,$$

其中 ξ 是 s 的连续函数．因为$\binom{-s}{k}$在 $[0,1]$ 中不改变符号，我们看到依第二中值定理，$L_n(y(t_n))$ 取形式

$$(-1)^k h^{k+1}\int_0^1\binom{-s}{k}y^{(k+1)}(\xi(s))ds$$

$$= (-1)^k h^{k+1} y^{(k+1)}(\xi) \int_0^1 \binom{-s}{k} ds$$

$$= \gamma_k h^{k+1} y^{(k+1)}(\xi) \qquad \xi \in (t_{n-1}, t_n), \tag{7.18}$$

这是误差的一个精确表示式，它可以转换成界 (7.16). 对于形式为 (7.15) 的一般方程，我们总可以得到一个类似于 (7.16) 的误差界. 可惜的是象在 (7.18) 那样包含导数的误差精确表示式仅仅对于某些方法能够给出. 但是，通过在 Taylor 级数中带有一附加项和高阶余项的办法，我们总可以得到形式为

$$C_{k+1} h^{k+1} y_n^{(k+1)} + O(h^{k+2})$$

的误差项. 在一般情况下，从

$$C_{k+1} = \frac{L_h(t^{k+1})}{h^{k+1}(k+1)!} \tag{7.19}$$

C_{k+1}能够容易得到.

7.3. 隐式多步方法——Adams-Moulton 方法

我们取等式 (7.2) 和 (7.3) 中的向量 $\mathbf{c}$ 为 $[\beta_0^*, 0, \cdots, 0, 1, 0, \cdots, 0]^T$，其中 1 出现在第 k 个位置上，由 (7.2)，

$$y_{n,(1)} = y_{n,(0)} + \beta_0^*(hf(y_{n,(0)}) - hy'_{n,(0)})$$

$$= \sum_{i=1}^{k} [(\alpha_i - \beta_0^* \gamma_i) y_{n-i} + (\beta_i - \beta_0^* \delta_i) h y'_{n-i}]$$

$$+ \beta_0^* hf(y_{n,(0)})$$

或

$$y_{n,(1)} = \sum_{i=1}^{k} (\alpha_i^* y_{n-i} + \beta_i^* h y'_{n-i}) + \beta_0^* hf(y_{n,(0)}), \tag{7.20}$$

其中

$$\alpha_i^* = \alpha_i - \beta_0^* \gamma_i, \qquad \beta_i^* = \beta_i - \beta_0^* \delta_i.$$

因此，由 (7.3) 和 (7.20) 得到

$$
\begin{aligned}
hy'_{n,(m+1)} &= hf(y_{n,(m)}) \\
y_{n,(m+1)} &= y_{n,(m)} + \beta_0^*(hf(y_{n,(m)}) - hy'_{n,(m)}) \\
&= y_{n,(1)} + \beta_0^*(hf(y_{n,(m)}) - hy'_{n,(1)}) \qquad (7.21) \\
&= \sum_{i=1}^{k}(\alpha_i^* y_{n-i} + \beta_i^* hy'_{n-i}) + \beta_0^* hf(y_{n,(m)}).
\end{aligned}
$$

如果(7.21)迭代到收敛，我们得到

$$
\begin{aligned}
y_n &= \sum_{i=1}^{k}(\alpha_i^* y_{n-i} + \beta_i^* hy'_{n-i}) + \beta_0^* hf(y_n), \\
hy'_n &= hf(y_n). \qquad (7.22)
\end{aligned}
$$

方程(7.22)定义了一个隐式的多步方法．它是隐式的，因为方程(7.22)由于含有函数 f 而是一个非线性方程，并且必须对 y_n 求解．预估-校正过程(7.1)和(7.3)是一个求解的办法．Adams-Moulton 方法［Moulton (1926)］是隐式方法的一个特例。

可以用许多方法来推导它．最简单的方法是在(7.6)中令 $m=n$ 且代入(7.5)得到

$$
y(t_n) = y(t_{n-1}) + \int_{t_{n-1}}^{t_n}\left[\sum_{j=0}^{k-1}(-1)^j\binom{-s+1}{j}\nabla^j f_n + (-1)^k h^k\binom{-s+1}{k}y^{(k+1)}(\xi)\right]ds.
$$

由此我们得到方法

$$
y_n = y_{n-1} + h\sum_{j=0}^{k-1}\gamma_j^*\nabla^j f_n, \qquad (7.23)
$$

其中

$$
\gamma_j^* = (-1)^j\int_0^1\binom{-s+1}{j}ds.
$$

利用第二中值定理，误差项是

$$
h^{k+1}(-1)^k\int_0^1\binom{-s+1}{k}y^{(k+1)}(\xi)ds = \gamma_k^* h^{k+1}y^{(k+1)}(\xi).
$$

用 $f_n, f_{n-1}, f_{n-2}, \cdots$ 代替 $\nabla^i f_n$，我们得到

$$y_n = y_{n-1} + h\sum_{i=0}^{k-1} \beta_{ki}^* f_{n-i},$$

其中

$$\beta_{ki}^* = (-1)^i \sum_{j=i}^{k-1} \binom{j}{i} \gamma_j^*.$$

Henrici (1962) §5.1.2 对于 γ_j^* （它可以利用生成函数得到）和 β_{ki}^* 给出了下面的值：

表 7.4 Adams-Moulton 方法的系数

m	0	1	2	3	4	5
γ_m^*	1	$-\frac{1}{2}$	$-\frac{1}{12}$	$-\frac{1}{24}$	$-\frac{19}{720}$	$-\frac{3}{160}$

表 7.5 Adams-Moulton 方法的系数

i	0	1	2	3	4	5
β_{1i}^*	1					
$2\beta_{2i}^*$	1	1				
$12\beta_{3i}^*$	5	8	−1			
$24\beta_{4i}^*$	9	19	−5	1		
$720\beta_{5i}^*$	251	646	−264	106	−19	
$1440\beta_{6i}^*$	475	1427	−798	482	−173	27

在 Adams-Bashforth 和 Adams-Moulton 方法之间存在三个要注意的重要差别．第一是后者的系数较小．这不只是导至较小的舍入误差，而且更重要的是在同一阶有较小的截断误差，因为 γ_i^* 比 γ_i 小且截断误差分别是 $\gamma_k^* h^{k+1} y^{(k+1)}$ 和 $\gamma_k^* h^{k+1} y^{(k+1)}$．第二个差别是对同样的阶，Adams-Moulton 方法利用较少的点的信息．换句话说，k 步 Adams-Bashforth 方法是 k 阶的，而 k 步的 Adams-Moulton 方法则是 $k+1$ 阶的．为

了推导这些公式，我们考虑待定系数方法时，这一点的理由是明显的。我们试验公式

$$y_n = \alpha_1^* y_{n-1} + h \sum_{i=0}^{k} \beta_i^* f_{n-i}, \tag{7.24}$$

这里共有 $k+2$ 个未知数，所以，能够使 Taylor 级数一致到 h^{k+1} 项而得到 $k+1$ 阶的 Adams-Moulton 方法。对于 Adams-Bashforth 方法，取 $\beta_0^* = 0$，未知数个数减 1，所以，仅能达到 k 阶。

7.4. 预估-校正方法

求解隐式方程 (7.22) 的最通常的方法是预估-校正过程 (7.1)，(7.2) 和 (7.3)。当使用多步方法时，它们是

$$y_{n,(0)} = \sum_{i=1}^{k} (\alpha_i y_{n-i} + \beta_i h y'_{n-i})$$

和

$$y_{n,(m+1)} = \sum_{i=1}^{k} (\alpha_i^* y_{n-i} + \beta_i^* h y'_{n-i}) + \beta_0^* h f(y_{n,(m)}). \tag{7.25}$$

如果从 (7.25) 减去 (7.22)，若 f 具有对 y 的连续偏导数，我们得到

$$\begin{aligned} y_{n,(m+1)} - y_n &= \beta_0^* h(f(y_{n,(m)}) - f(y_n)) \\ &= \beta_0^* h \frac{\partial f(\xi_n)}{\partial y} (y_{n,(m)} - y_n). \end{aligned}$$

因此，在我们感兴趣的区域内，如果 h 取得充分小，使得 $|h\beta_0(\partial f/\partial y)| < 1$，则 $|y_{n,(m+1)} - y_n| < |y_{n,(m)} - y_n|$ 且迭代 (7.25) 将收敛到 (7.22) 的解。实际上，预估 $y_{n,(0)}$ 是一个很好的近似，所以，要求很少几次（二或三次）校正。

预估公式是 7.2 所讨论的类型的一个显式多步方法。所谓它的阶，意指当 y_{n-1} 为精确值时，$y_{n,(0)}$ 对 y_n 的近似的阶。

校正公式的阶是当校正迭代到收敛时方法的阶，即它是方法(7.22)的阶。

预估和校正公式不必是同阶的.每一次应用附加的校正，在达到校正之前，将把解的阶加1. 因此，如果预估有阶 q 且校正有阶 r，我们就有

$$y(t_n) = \sum_{i=1}^{k} [\alpha_i y(t_{n-i}) + h\beta_i y'(t_{n-i})] + O(h^{q+1})$$

$$= y_{n,(0)} + O(h^{q+1}) \qquad \text{(预估公式)},$$

$$y(t_n) = \sum_{i=1}^{k} [\alpha_i^* y(t_{n-i}) + \beta_i^* h y'(t_{n-i})]$$

$$+ \beta_0^* h f(y(t_n)) + O(h^{r+1})$$

$$= y_{n,(1)} + h\beta_0^* [f(y(t_n)) - f(y_{n,(0)})] + O(h^{r+1})$$

$$= y_{n,(1)} + O(h^{q+2}) + O(h^{r+1}) \text{(第一次应用校正公式)}.$$

考虑到应用 m 次校正，

$$y(t_n) = y_{n,(m)} + O(h^{q+m+1}) + O(h^{r+1}).$$

但是，实际上我们不希望校正许多次，因为每一次附加的校正步对于导数要求一个附加的函数计算. 通常 q 取作 $r-1$ 或 r 且大约使用两次校正步[见 Hull 和 Creemer (1963)].

会出现三个独立的过程：即预估步，我们称为 P；根据 y 的最后值的导数求值称为 E；校正步称为 C. 有一个用 C 步还是用 E 步来结束的选取，有一些理由证明，用求值结束是优越的. 这将在第 8 章和第 9 章中讨论.一个 m 次迭代的预估-校正方法称为 $P(EC)^m$ 方法，如果它用校正结束，或者称为 $P(EC)^m E$ 方法，如果它用函数求值结束.整个方法类称为 PC 方法类.

问　　题

1. 证明等式(7.19)前面的命题

2. 推导 Adams-Moulton 方法的系数生成函数。

3. (a) 用求 $\bar{\gamma}_i$ 的生成函数的办法，推导方法

$$y_n = y_{n-1} + h\sum_{i=0}^{k}\bar{\gamma}_i\nabla^i f_{n-2}$$

的一般形式。

(b) 截断误差是什么？

4. (a) 推导方法

$$y_n = y_{n-1} + h\sum_{i=0}^{k-1}\bar{\gamma}_i\nabla^i f_{n+1}$$

的一般形式和截断误差。

(b) 你能够提出使用这个方法的任何途径吗？

5. 用下面提到的每一种方法，使用计算机从 $y(0)=0$ 到 $t=1$ 用 $h=2^{-k}(k=1,2,\cdots,5)$ 积分微分方程

$$y' = t^3 - y + 3t^2.$$

(a) 一、二和三步的 Adams-Bashforth 方法；

(b) 一和二步 $P(EC)^1$ Adams-Bashforth-Moulton 方法（校正阶比预估的阶大 1）；

(c) 一和二步 Adams-Moulton 方法（对 y_n 显式解出，因为 f 关于 y 是线性的）。

打印在 $t=1$ 的误差且说明之。

8. 一般的多步方法、阶和稳定性

在前一章,我们研究了两种特殊的多步方法,它们对于积分许多微分方程是很有效的. 本章,我们要研究形如

$$\sum_{i=0}^{k}(\alpha_i y_{n-i} + h\beta_i f_{n-i}) = 0 \tag{8.1}$$

的一般的 k 步方法,并且讨论稳定性这个重要的问题.

使我们能够推导 Adams 方法的系数的一种办法,是要求它们对于阶 $\leqslant r$ 的多项式是精确的. 在 (8.1) 中有 $2k+2$ 个未知数,有一个任意的标准化因子,所以,我们可令 $\alpha_0 = -1$. 于是,余下 $2k+1$ 个未知数. 因此,我们指望能选择 α 和 β,使得这方法对于阶高达 $2k$ 的多项式是精确的. 这是可能的[1].

---

1) 假定已知 $y_{n-k}, \cdots, y_n, y'_{n-k}, \cdots, y'_n$. 存在一个次数为 $2k+1$ 的唯一的多项式通过这函数,并且在 $t = t_{n-k}, \cdots, t_n$ 与函数的导数一致. 这称为 Hermite 内插公式,并按如下的方式给出. 令

$$\phi(\xi) = (\xi - t_{n-k})\cdots(\xi - t_n)$$

及

$$\phi_i(\xi) = \frac{\phi(\xi)}{\xi - t_{n-i}},$$

那么

$$P(\xi) = \sum_{j=0}^{k} \frac{\phi_i(\xi)}{\phi_i^2(t_{n-i})}\left[\phi_i(\xi) y_{n-i} - \phi(\xi)\Big(y'_{n-i} - 2\sum_{j, j\neq i}\frac{y_{n-i}}{t_{n-i} - t_{n-j}}\Big)\right]$$

是所要求的多项式, 这由代入可以看出. 如果 ξ^{2k+1} 的系数为零,则有一个次数为 $2k$ 的多项式通过诸点 y_{n-i} 及其导数 y'_{n-i}. 系数为零的条件是使 y_n 和 y'_n 的值与其他的 y_{n-i} 和 y'_{n-i} 的值相关的条件. 如果

$$-\sum_{i=0}^{k}\frac{1}{\phi_i^2(t_{n-i})}\Big(y'_{n-i} - 2\sum_{j, j\neq i}\frac{y_{n-i}}{t_{n-i} - t_{n-j}}\Big) = 0,$$

则 ξ^{2k+1} 的系数为零. 如果令

$$\alpha_i = \frac{2}{\phi_i^2(t_{n-i})}\sum_{j, j\neq i}\frac{1}{t_{n-i} - t_{n-j}}$$

但是我们以后会看到这样的方法，对于 $k>2$ 是毫无用处的，当 $k=2$ 时，仅仅在某种程度上有用. 如果我们只涉及到局部截断误差而且问题又具有变化性质好的导数，则总是想用最大阶为 $2k$ 的 k 步方法. 可是，我们以后会看到，对于 $k>2$，这样的方法由于不稳定性的缘故，使得在一步中产生的小的截断误差在后面的各步中被放大到不可接受的程度. 但是，却存在阶为 $k+1$（例如，Adams-Moulton 方法）以及若 k 是偶数阶为 $k+2$ 的稳定的 k 步方法.

我们已经讨论了使方程（8.1）对于各种不同阶的多项式为精确的问题，在上一章称它为方法的阶. 下节我们讨论多步方法的阶的含义，然后转向最大阶二步方法的分析来了解稳定性问题. 本章最后考察三步方法类，以便了解精确度和稳定性两者间折衷的可能性.

8.1. 多步方法的阶

定义

$$L_h(y(t))=\sum_{i=0}^{k}(\alpha_i y(t-h_i)+h\beta_i y'(t-h_i)).$$

及

$$\beta_i=\frac{1}{\phi_i^2(t_{n-i})h},$$

则这些与(8.1)的左端等价. 对于这些 α 和 β 的值，次数为 $2k$ 或小于 $2k$ 的任意多项式都满足(8.1). 必须证明 $\alpha_0\neq 0$. 这是显然的，因为 $\phi_0(t_n)\neq 0$，又 $1/(t_n-t_{n-j})>0$，对于 $j\geq 1$. 用 $h^{2k+1}(k!)^2$ 乘 α 和 β，又令 $t_{n-i}-t_{n-j}=h(j-i)$，我们得到

$$\alpha_i=\begin{cases}2\binom{k}{i}^2\sum\limits_{j=i+1}^{k-i}\frac{1}{j} & i\leq\frac{k}{2},\\ -2\binom{k}{i}^2\sum\limits_{j=k-i+1}^{i}\frac{1}{j} & i\geq\frac{k}{2};\end{cases}\tag{8.2}$$

$$\beta_i=-\binom{k}{i}^2.$$

在第 7 章我们知道能够选取 α 和 β 使得关于 y 的 Taylor 级数的若干项为零.

定义 8.1. 算子 L_h 的阶为满足下述条件的 r 的最大数：如果 $y(t)$ 有连续的 $r+1$ 阶导数,则

$$L_h(y(t)) = O(h^{r+1}). \tag{8.3}$$

若假定 y 有连续的 $r+2$ 阶导数,则能够用余项为 $O(h^{r+2})$ 的 Taylor 级数代替 y 和 y'. 如果按 $h^0, h^1, h^2, \cdots, h^{r+1}$ 诸项进行整理,则得

$$L_h(y(t)) = \sum_{q=0}^{r+1} C_q h^q y^{(q)}(t) + O(h^{r+1}),$$

其中

$$C_q = \begin{cases} \sum_{i=0}^{k} \alpha_i & q = 0, \\ \sum_{i=0}^{k} \left[\frac{(-i)^q}{q!} \alpha_i + \frac{(-i)^{q-1}}{(q-1)!} \beta_i \right] & q > 0. \end{cases} \tag{8.4}$$

线性方程 $C_q = 0 (q \leqslant r)$ 是确定 r 阶方法的方程. 我们注意 C_{r+1} 仅仅依赖于方法的系数,不依赖于 y 或展开点 t. C_{r+1} 是 L_h 的截断误差系数. 方程 (8.1) 以及 L_h 可以乘上任意常数. 在第 10 章将证明,每步所引进的截断误差总和是

$$\frac{C_{r+1}}{\sum_{i=0}^{k} \beta_i} h^{r+1} y^{(r+1)} + O(h^{r+2}).$$

因此,很自然比例因子取成使

$$\sum_{i=0}^{k} \beta_i = 1. \tag{8.5}$$

我们将假定当讨论误差系数 C_{r+1} 时,β_i 已是标准化的了.

如果使用阶为 r 的方法来积分一个方程, 其解是次数不超过 r 的多项式,则解将除了舍入误差外是精确的,因为截断

误差是 $\sum_{r+1}^{\infty} C_q h^q y^{(q)}(t) = 0$. 但是，看看表 8.6 就可说明在某些情形舍入误差可以破坏这个解. 阶只是告诉我们局部截断误差如何按照 h 的函数来变化，而系数 C_{r+1} 却给我们一个方法来比较具有同样阶的两个不同方法中的误差.

阶 r 和误差系数 C_{r+1} 可以按如下更方便的方式来表达. 定义多项式

$$\rho(\xi) = \sum_{i=0}^{k} \alpha_i \xi^{k-i},$$

$$\sigma(\xi) = \sum_{i=0}^{k} \beta_i \xi^{k-i}, \tag{8.6}$$

ρ 和 σ 最大次数是方法的步数. 通常 σ 的次数 $\leqslant \rho$ 的次数. 如果严格的不等式成立，则方法是显式的. 考虑函数 $y(t) = e^{\lambda t}$,

$$\begin{aligned} L_h(y(t)) &= \sum_{i=0}^{k} (\alpha_i + \lambda h \beta_i) e^{\lambda(t-ht)} \\ &= \sum_{i=0}^{k} e^{\lambda(t-hk)} (\alpha_i + h\lambda\beta_i)(e^{h\lambda})^{k-i} \\ &= e^{\lambda(t-hk)} [\rho(e^{h\lambda}) + h\lambda\sigma(e^{h\lambda})]. \end{aligned} \tag{8.7}$$

因方法是 r 阶的，又因 $e^{hk} = 1 + O(h)$, 所以

$$\begin{aligned} L_h(y(t)) &= C_{r+1} h^{r+1} y^{(r+1)} + O(h^{r+2}) \\ &= C_{r+1} (h\lambda)^{r+1} e^{\lambda t} + O(h^{r+2}) \\ &= C_{r+1} (h\lambda)^{r+1} e^{\lambda(t-hk)} + O(h^{r+2}). \end{aligned} \tag{8.8}$$

因此，由 (8.7) 及 (8.8), 得

$$\rho(e^{h\lambda}) + h\lambda\sigma(e^{h\lambda}) = C_{r+1}(h\lambda)^{r+1} + O(h^{r+2}). \tag{8.9}$$

记 $h\lambda = \log(1+z)$ 并注意 $h\lambda = z + O(z^2)$, 于是

$$\rho(1+z) + \log(1+z)\sigma(1+z) = C_{r+1} z^{r+1} + O(z^{r+2}). \tag{8.10}$$

从而方程 (8.10) 是方法具有阶为 r 的一个必要条件. 它也能

看成是一充分条件，只要注意，由它可得

$$L_h(e^{\lambda t}) = \sum_{q=0}^{\infty} C_q(h\lambda)^q e^{\lambda t} = O(h^{r+1}),$$

这又意味着 $C_0 = C_1 = \cdots = C_r = 0$.

如果方程标准化使得 (8.5) 是真确的，则 $\sigma(1) = 1$. 展开 (8.10) 为 z 的幂级数，我们得到

$$\begin{aligned}\rho(1) + z\rho'(1) + O(z^2) + [z + O(z^2)][\sigma(1) + O(z)] \\ = \rho(1) + z[\rho'(1) + \sigma(1)] + O(z^2) \\ = C_{r+1}z^{r+1} + O(z^{r+2}).\end{aligned}$$

由此得知，由阶≥0 可推出 $\rho(1) = 0$，由阶≥1 可推出$\rho'(1) + \sigma(1) = 0$.

8.1.1. 给定 α, β 的一个确定另一个

如果给定多项式 σ，则方程 (8.10) 表示如何能求得次数为 k 的唯一多项式$\rho(\xi)$使方法具有阶 $\geqslant k$. 令 $r(z)$为由 $\log(1+z)\cdot\sigma(1+z)$ 展式中 $z^j(0 \leqslant j \leqslant k)$ 的项来得到这个多项式. 因此，给出 $\rho(\xi)$ 为 $r(\xi - 1)$.

例.

如果 $\sigma(\xi) = \frac{3}{2}\xi - \frac{1}{2}$, $k = 2$，则得

$$\begin{aligned}\rho(1+z) &= -\log(1+z)\left(\frac{3}{2}(1+z) - \frac{1}{2}\right) + O(z^3) \\ &= -\left(z - \frac{z^2}{2}\right)\left(\frac{3}{2}z + 1\right) + O(z^3) \\ &= -z - z^2 = -(1+z)^2 + (z+1).\end{aligned}$$

于是，$\rho(\xi) = -\xi^2 + \xi$，这给出一个二阶 Adams-Bashforth 方法

$$-y_n + y_{n-1} + \frac{3h}{2}y'_{n-1} - \frac{h}{2}y'_{n-2} = 0.$$

相反，如果 $\rho(\xi)$ 给定，则存在次数为 k 的 $\sigma(\xi)$，使得这方法的阶 $\geqslant k+1$.在 $r=k+1$ 的情形，用 $\log(1+z)$除(8.10)，得到

$$\sigma(1+z)=\frac{z}{\log(1+z)}\left[-\frac{\rho(1+z)}{z}+C_{k+2}z^{k+1}+O(z^{k+2})\right].$$

因为 $z/\log(1+z)$ 在 $z=0$ 是解析的，而且若 $k\geqslant 0$，$\rho(1+z)/z$ 在 $z=0$ 附近也必定是解析的．这就是说，

$$\rho(1)=\sum_{j=0}^{k}\alpha_i=0.$$

所以，通过如下方式我们能求 σ：即令 $s(z)$ 等于

$$-\frac{z}{\log(1+z)}\cdot\frac{\rho(1+z)}{z}$$

的展开式的 $z^j(0\leqslant j\leqslant k-1)$ 项，而且记 $\sigma(\xi)=s(\xi-1)$.

例.

如果 $\rho(\xi)=-\xi^2+\xi, k=2$，则得

$$\begin{aligned}\sigma(1+z)&=\frac{(1+z)^2-(1+z)}{\log(1+z)}+O(z^3)\\&=\frac{1+z}{1-\frac{z}{2}+\frac{z^2}{3}}+O(z^3)\\&=(1+z)\left(1+\frac{z}{2}-\frac{z^2}{3}+\frac{z^2}{4}\right)+O(z^3)\\&=1+\frac{3z}{2}+\frac{5z^2}{12}\\&=\frac{5}{12}(1+z)^2+\frac{2}{3}(1+z)-\frac{1}{12}.\end{aligned}$$

于是，$\sigma(\xi)=\frac{5}{12}\xi^2+\frac{2}{3}\xi-\frac{1}{12}$，这就给出三阶 Adams-Moulton 方法：

$$-y_n + y_{n-1} + \frac{5}{12}hy'_n + \frac{2}{3}hy'_{n-1} - \frac{1}{12}hy'_{n-2}.$$

8.1.2. 方法的主根

如果把多步方法应用到 $y' = \lambda y$, 则得递推关系式

$$\sum_{i=0}^{k}(\alpha_i + h\lambda\beta_i)y_{n-i} = 0.$$

这种形式的方程的通解将在第10章详细讨论.这里只要注意:如果 ξ 是方程

$$\sum(\alpha_i + h\lambda\beta_i)\xi^{k-i} = \rho(\xi) + h\lambda\sigma(\xi) = 0$$

的一个根，我们能够找到形如 $y_n = A\xi^n$ 的解. 由于 $y' = \lambda y$ 的解是 $y = Ae^{\lambda t} = A(e^{h\lambda})^n$,我们期望一个根为 $e^{h\lambda}$ 的近似，使得 y_n 能够近似 $y(t_n)$. 我们称这个根为主根(principal root) ξ_1，其他根 ξ_i 也产生解 $A\xi_i^n$，这些根称为"附加"根 (extraneous roots)."附加"根的大小影响方法的稳定性，这将在下节通过一个例子来讨论.

可以证明,如果 $\rho'(1) \neq 0$, 则主根是

$$\xi_1 = e^{h\lambda} - \frac{C_{r+1}}{\rho'(1)}(h\lambda)^{r+1} + O(h^{r+2}). \qquad (8.11)$$

设有一个形如 $e^{h\lambda} + \gamma$ 的根,其中 γ 待确定,我们有

$$\begin{aligned}0 &= \rho(e^{h\lambda} + \gamma) + h\lambda\sigma(e^{h\lambda} + \gamma)\\ &= \rho(e^{h\lambda}) + \gamma\rho'(e^{h\lambda}) + h\lambda\sigma(e^{h\lambda}) + O(\gamma^2 + \gamma h).\end{aligned}$$

考虑到(8.9),我们可把它写成

$$0 = C_{r+1}(h\lambda)^{r+1} + \gamma\rho'(e^{h\lambda}) + O(h^{r+2}) + O(\gamma^2 + \gamma h).$$

因为 $\rho'(e^{h\lambda}) = \rho'(1 + O(h)) = \rho'(1) + O(h)$, 由此, 如果 $\rho'(1) \neq 0$, 则得出形如(8.11)的主根. 若 $r \geq 1$, 已在§ 8.1 末尾证明过 $\rho'(1) + \sigma(1) = 0$. 为了标准化, 我们已经要求 $\sigma(1) = 1$,于是,最后可以写

$$\xi_1 = e^{h\lambda} + C_{r+1}(h\lambda)^{r+1} + O(h^{r+2}) \qquad (8.12)$$

8.2. Milne 方法

为了得到最大可能阶的二步方法，我们要求 $C_0 = C_1 = C_2 = C_3 = C_4 = 0$ 或由 (8.4) 得

$$\alpha_0 + \alpha_1 + \alpha_2 = 0,$$
$$-\alpha_1 - 2\alpha_2 + \beta_0 + \beta_1 + \beta_2 = 0,$$
$$\frac{\alpha_1}{2!} + \frac{4\alpha_2}{2!} - \beta_1 - 2\beta_2 = 0,$$
$$-\frac{\alpha_1}{3!} - \frac{8\alpha_2}{3!} + \frac{\beta_1}{2!} + \frac{4\beta_2}{2!} = 0,$$
$$\frac{\alpha_1}{4!} + \frac{16\alpha_2}{4} - \frac{\beta_1}{3!} - \frac{8\beta_2}{3!} = 0.$$

这些方程有解 $\alpha_0 = -\alpha_2 = -1$，$\alpha_1 = 0$，$\beta_0 = \beta_2 = \frac{1}{3}$，$\beta_3 =$

表 8.1. $y' = y$ 用 Milne 方法的解

时间	y	误差
0.0	0.1000000000E 01	0.0
0.1	0.1105171204E 01	0.0
0.2	0.1221402168E 01	−0.95367E-06
0.3	0.1349857330E 01	−0.95367E-06
0.4	0.1491822243E 01	−0.19073E-05
0.5	0.1648717880E 01	−0.28610E-05
0.6	0.1822114944E 01	−0.38147E-05
0.7	0.2013747215E 01	−0.47684E-05
0.8	0.2225535393E 01	−0.47684E-05
0.9	0.2459595680E 01	−0.66757E-05
1.0	0.2718274117E 01	−0.66757E-05
2.0	0.7389023781E 01	0.95367E-05
3.0	0.2008541870E 02	0.10681E-03
4.0	0.5459768677E 02	0.47302E-03
5.0	0.1484117584E 03	0.19989E-02
6.0	0.4034238281E 03	0.65918E-02
7.0	0.1096617187E 04	0.21729E-01
8.0	0.2980913330E 04	0.74707E-01
9.0	0.8102945312E 04	0.23047E 00
10.0	0.2202604687E 05	0.71484E 00

$\frac{4}{3}$，由此得出

$$y_n = y_{n-2} + \frac{1}{3} h(f_n + 4f_{n-1} + f_{n-2}). \tag{8.13}$$

这就是通常所说的 Milne 方法，而且可看成是与 Simpson 求积法则类似的方法. 在下面的例子中，用 Milne 方法利用步长 $h = 0.1$，从 $y(0) = 1$ 到 $t = 10$ 积分 $y' = y$. 对 $t = 0(0.1)1$ 和 $t = 2(1)10$，计算结果和误差已列于表 8.1.对于较小的计算量来说，精确度是好的($y(0.1)$的值由指数子程序计算. 在 IBM/360 机器上按单精度完成，大约给出七位十进位数字). 与此相反，表 8.2 列出了从 $y(0) = 1$ 到 $t = 10$ 用同样的步长积分方程 $y' = -y$ 的类似结果，其最后答案仅保留一位十进位数字的精度，而对于 $y' = y$ 的精度，比四位十进位数字还要好.显然，这个特殊方法对第二个问题不那么合适. 为了看

表 8.2. 用 Milne 方法求解 $y' = -y$ 的解

时间	y	误差
0.0	0.1000000000E 01	0.0
0.1	0.9048374295E 00	0.0
0.2	0.8187311888E 00	0.35763E-06
0.3	0.7408185601E 00	0.23842E-06
0.4	0.6703208089E 00	0.65565E-06
0.5	0.6065312028E 00	0.41723E-06
0.6	0.5488125086E 00	0.77486E-06
0.7	0.4965859056E 00	0.47684E-06
0.8	0.4493299127E 00	0.83447E-06
0.9	0.4065702558E 00	0.47684E-06
1.0	0.3678804040E 00	0.83447E-06
2.0	0.1353359818E 00	−0.59605E-07
3.0	0.4978756607E-01	−0.70781E-07
4.0	0.1831618324E-01	0.23097E-06
5.0	0.6738595665E-02	0.49546E-06
6.0	0.2479620045E-02	0.79698E-06
7.0	0.9130770341E-03	0.11637E-05
8.0	0.3371228082E-03	0.16459E-05
9.0	0.1257221593E-03	0.23067E-05
10.0	0.4862506466E-04	0.32228E-05

看选取特别坏的步长的情形，用 $h=10^{-i}(i=1,2,\cdots,5)$ 积分同一问题，结果在表 8.3 列出.

当步长缩小，误差无疑不像所期望的那样按 h^4 减小. 显然，即使积分很少几步，舍入误差也是不可忽视的.

表 8.3. 对 $y'=-y, y(1)=1$ 用 Milne 方法求得的 $y(10)$ 的值

h	y	误差
0.10000E 00	0.48625E-04	−0.32251E-05
0.10000E-01	0.48211E-04	−0.28114E-05
0.10000E-02	0.38639E-04	0.67607E-05
0.10000E-03	0.32934E-04	−0.37534E-04
0.10000E-04	0.57846E-04	−0.12446E-04

8.2.1. 对于 $y'=\lambda y$ Milne 方法的稳定性

显然，Milne 方法的误差当 $\lambda=-1$ 时是增长的，这样自然认为是稳定性问题. 我们来研究在一步上对数值解的扰动带给以后诸步的影响. 在 (8.13) 中令 $f=\lambda y$，并考虑 (8.13) 的两个不同解之间的差 e_n，我们得到

$$e_n=e_{n-2}+\frac{1}{3}\lambda h(e_n+4e_{n-1}+e_{n-2}). \qquad (8.14)$$

这是关于 e_n 的一个二阶齐次线性差分方程. 通常这种方程可通过寻找形如 $e_n=\xi^n$ 的解来解决. 代入 (8.14)，我们得到

$$\left(1-\frac{1}{3}h\lambda\right)\xi^2-\frac{4}{3}h\lambda\xi-\left(1+\frac{1}{3}h\lambda\right)=0. \qquad (8.15)$$

注意，这个方程恰好是 $\rho(\xi)+h\lambda\sigma(\xi)=0$. 如果这方程有两个不同的根 ξ_1 和 ξ_2，则形如 $a\xi_1^n+b\xi_2^n$ 的 e_n 能够看成是 (8.14) 的解. 如果 $1\pm\frac{1}{3}h\lambda\neq 0$，则固定任意两个相邻的 e_i 和 e_{i+1}，就可以唯一确定一切其他的 e_i. 于是 (8.14) 的全部解能用

$a\xi_1^n + b\xi_2^n$ 来表示（总能选取 a 和 b 使得 $a\xi_1^i + b\xi_2^i = e_i$, $a\xi_1^{i+1} + b\xi_2^{i+1} = e_{i+1}$）. 如果 $\xi_1 = \xi_2$,则 (8.14) 的通解可写成 $(a + bn)\xi_1^n$.

我们立刻会知道,如果两个中的任何一个 $|\xi_i| > 1$,则扰动按指数增长;如果 $\xi_1 = \xi_2$, 且 $|\xi_1| = 1$, 则扰动线性增长;而如果都是 $|\xi_i| < 1$, 则扰动减小. 方程 (8.15) 可用待定系数方法[1]对 ξ 求形为 $h\lambda$ 的幂级数的解,得到

$$\xi_1 = 1 + h\lambda + \frac{(h\lambda)^2}{2} + \frac{(h\lambda)^3}{6} + \frac{(h\lambda)^4}{24} + \frac{(h\lambda)^5}{72} + O(h^6)$$

$$= e^{h\lambda} + \frac{1}{180}(h\lambda)^5 + O(h^6),$$

$$\xi_2 = -\left[1 - \frac{h\lambda}{3} + \frac{(h\lambda)^2}{18} + \frac{5(h\lambda)^3}{54} + O(h^4)\right]$$

$$= -e^{-h\lambda/3} + O(h^3).$$

根 ξ_1 与 $e^{h\lambda}$ 一致到 $O(h^5)$ 是预期的, 因为所计算的解按四阶方法将接近真解 $e^{h\lambda}$ 到 $O(h^5)$.所引进的任何误差都会有将解变到解族中的另一个解上的效应, 如图 1.1 所示. 因此,我们期望所引进的误差的某些部分按 $e^{\lambda t} = (e^{h\lambda})^n \cong \xi_1^n$ 变化.

第二个根 ξ_2 的出现是一种在单步方法中不会发生的现象,这是由于贮存了"附加的信息"(additional information)而产生的. 对于 $\lambda > 0$, 解的第二个分量 $\xi_2^n \cong (-1)^n e^{-\lambda t/3}$ 是衰减的,于是不出现问题的. 然而,如果 $\lambda < 0$,则第二个分量将超过第一个分量和真解.根据第 1 章绝对稳定性的定义,我们知

---

1) 令 $\xi_1 = a_0 + a_1 h\lambda + a_2(h\lambda)^2 + \cdots$,
$\xi_2 = b_0 + b_1 h\lambda + b_2(h\lambda)^2 + \cdots$,
并且按照

$$\left(1 - \frac{h\lambda}{3}\right)(\xi - \xi_1)(\xi - \xi_2) = \left(1 - \frac{h\lambda}{3}\right)\xi^2 - \frac{4h\lambda}{3}\xi + \left(1 + \frac{h\lambda}{3}\right)$$

使得 $(h\lambda)^n$ 的系数相等.

道 Milne 方法是绝对不稳定的，除非 $\mathrm{Re}(\lambda)=0$. 但是，当 $\lambda>0$ 时，这种不稳定性是由于问题本身的不稳定性所引起的，而且是不严重的. 反之，如果 $\lambda<0$，不稳定性则是由于计算方法产生的第二个分量引起的.

8.3. 一般的多步方法的稳定性

在第 1 章中，稳定性定义对于一切 $h\leqslant h_0$ 初始值中扰动的影响是有界的. 如果我们考虑对线性问题 $y'=\lambda y+f(t)$ 数值解的扰动，由方程 (8.1) 得到一个形式为

$$\sum_{i=0}^{k}(\alpha_i+h\lambda\beta_i)e_{n-i}=0 \tag{8.16}$$

的误差方程. 我们再来寻求形为 $e_n=\xi^n$ 的解，于是得到

$$\sum_{i=0}^{k}(\alpha_i+h\lambda\beta_i)\xi^{k-i}=0,$$

ξ 就是

$$\rho(\xi)+h\lambda\sigma(\xi)=0 \tag{8.17}$$

的根. 如果全部根 $\xi_j(j=1,\cdots,k)$ 是不相同的，则 (8.16) 的通解能写成

$$e_n=\sum_{j=1}^{k}\gamma_j\xi_j^n. \tag{8.18}$$

如果一些根是相等的，则这个解就必须改变. 例如，如果 ξ_i 是 m 重根，则要出现项$(\gamma_i+\gamma_{i+1}n+\gamma_{i+2}n^2+\cdots+\gamma_{i+m-1}n^{m-1})\xi_i^m$.

显然，如果任何一个 $|\xi_i|$ 大于 1，则扰动就增长，而且我们说方法对于 $h\lambda$ 是绝对不稳定的. 如果我们考虑 t 的一个无限区域，则问题将是不稳定的.

(8.17) 的解是 $h\lambda$ 的函数，即 $\xi_i=\xi_i(h\lambda)$. 多项式的根是其系数的连续函数. 因此，如果 $|\xi_i(0)|<1$，则对于固定的 λ，存在一个 h_0，使得当 $h\leqslant h_0$ 时 $|\xi_i(h\lambda)|\leqslant 1$. 如果有单重根 ξ_i 使 $|\xi_i(0)|=1$ [的确，如果阶 r 至少为零，主根 ξ_1 必须

是那种形式的，因 $\xi_1(h\lambda)=e^{h\lambda}+O(h^{r+1})$]，对于充分小的 h，则有 $\xi_1(h\lambda)=\xi_i(0)+O(h)$ [在 $h\lambda$ 的任何使 (8.18) 的根是各不相同的区域内，这些根都是 $h\lambda$ 的可微函数]. 因此，对于 $t\leqslant b$ 我们有

$$|\xi_i^n(h\lambda)|\leqslant|\xi_i(0)+kh|^n\leqslant(1+kh)^n\leqslant e^{khn}\leqslant e^{kt}\leqslant e^{kb}.$$

于是，我们觉得 $\rho(\xi)=0$ 的根必须服从一些这样的稳定性条件. 第 10 章证明：阶 $\geqslant 1$ 的多步方法的稳定性，其必要且充分条件是 $\rho(\xi)=0$ 的根在单位圆内，或者在单位圆上是单根.

这将称为"根条件"(root condition)而且我们把这样的 $\rho(\xi)$ 称为稳定多项式.

我们知道，在 Milne 方法中有多于一个单位圆上的根在某些问题中能引起有害的变化. 于是，我们定义如下术语：

定义 8.2. 如果 $\rho(\xi)=0$ 的全部根除了 $\xi=1$ 外均在单位圆内，则方法是强稳定的.

定义 8.3. 如果 $\rho(\xi)=0$ 在单位圆上有多于一个的根，而且其它根均在单位圆内，则称方法是弱稳定的.

不稳定公式极有害的性质在下面三阶公式的例子中可以看到：

$$y_n=-4y_{n-1}+5y_{n-2}+4hf_{n-1}+2hf_{n-2}.$$

这是最准确的二步显式公式. 如果以初始值 $y_0=0$, $y_1=\varepsilon$ (小的舍入误差!)来解方程 $y'=0$，我们得到表 8.4 的结果：

这个变化的性态是与 h 无关的，所以显然当 $h\to 0$ 时对于所计算的解收敛于真解是没有希望的.

第 10 章我们再正式定义收敛性(意指所计算的解通过把 h 选得足够小的办法能使其任意接近于真解) 而且证明稳定性以及阶 $\geqslant 1$ 是收敛性的必要且充分条件. 于是，我们知道

稳定性是那里的一个重要概念，它保证存在一种得到所要求的任何精确度的方法．但是，实际上我们计算用的是有限的 h，而且感兴趣的问题是：为了达到给定的精确度，需要多小的步长 h．这些问题常常涉及绝对稳定性概念，这点在下节讨论．

表 8.4. 不稳定性的影响

n	y
0	0
1	ϵ
2	-4ϵ
3	21ϵ
4	-104ϵ
5	521ϵ
6	-2604ϵ

8.3.1. 绝对稳定性

我们用三阶 Adams-Bashforth 方法，以步长 $h=\dfrac{1}{8}$ 来积分 $y'=\lambda(y-t^3)+3t^2$, $y(0)=0$．一种情况 $\lambda=-1$，另一种情况 $\lambda=-100$．在这两种情况下，解是 t^3．（对这个问题，假若没有舍入误差，三阶方法总是精确的．）

我们不考虑初始值计算的问题，因为我们知道

$$f_{-2}=y'\left(-\frac{1}{4}\right)=0.1875,$$

$$f_{-1}=y'\left(-\frac{1}{8}\right)=0.046875,$$

$$f_0=y'(0)=0,$$

$$y_0=y(0)=0.$$

当 $h=\dfrac{1}{8}$ 时三阶 Adams-Bashforth方法是

$$y_n=y_{n-1}+\frac{1}{96}(23f_{n-1}-16f_{n-2}+5f_{n-3}). \quad (8.19)$$

计算准确到正确舍入的八位十进制数字．对于 $\lambda=-1$，我们得到

表 8.5. $y' = t^3 - y + 3t^2$ 用 Adams-Bashforth 方法的解

$n = 8t$	y	误差 $= y - t^3$	y'	$\frac{y'}{96}$
−2			0.1875 0000	0.0019 5313
−1			0.0468 7500	0.0004 8828
0	0.0000 0000		0.0000 0000	0.0000 0000
1	0.0019 5317	+0.0000 0004	0.0468 7496	0.0004 8828
2	0.0156 2501	+0.0000 0001	0.1874 9999	0.0019 5312
3	0.0527 3429	−0.0000 0009	0.4218 7509	0.0043 9453
4	0.1249 9996	−0.0000 0004	0.7500 0004	0.0078 1250
5	0.2441 4058	−0.0000 0005	1.1718 7505	0.0122 0703
6	0.4218 7492	−0.0000 0008	1.6875 0808	0.0175 7813
7	0.6699 2193	+0.0000 0005	2.2968 7495	0.0239 2578
8	0.9999 9994	+0.0000 0006		

所引进的误差均为舍入误差. 注意在第一步引进的舍入误差 +0.00000004 的变化. 在计算 y' 时乘以 $\lambda = -1$,再在得到 $(h/12)y'$ 时除以96. 于是,它在舍入时就没有了. 所以, 下一个值 y_2 包含由 y_1 得来的同样的误差和加一个新的舍入误差, 在这种情况下为−0.00000003因为总的误差值是0.00000001. 继续这个过程, 舍入误差在每一步得到总的误差影响都小于 0.00000001 的平均值. 如果我们对于 $\lambda = -100$ 进行积分, 则得到表 8.6 的结果

表 8.6. $y' = 100t^3 - 100y + 3t^2$ 用 Adams-Bashforth 方法的解

$n = 8t$	y_n	误差 $= y_n - t^3$	y'	$\frac{y'}{96}$
−2			0.1875 0000	0.0019 5313
−1			0.0468 7500	0.0004 8828
0	0.0000 0000	0.0000 0000	0.0000 0000	0.0000 0000
1	0.0019 5317	+0.0000 0004	0.0468 7100	0.0004 8824
2	0.0156 2409	−0.0000 0091	0.1875 9100	0.0019 5407
3	0.0527 5568	+0.0000 2130	0.4197 4500	0.0043 7234
4	0.1244 9563	−0.0005 0437	0.8004 3700	0.0083 3789
5	0.2540 8001	+0.0099 3938	0.1779 3700	0.0018 5351
6	0.1851 6620	−0.2367 0880	25.3583 8000	0.2641 4979
7	6.2726 4466	+5.6027 2278		

注意这一次在y_1中同样的舍入误差的变化．在计算 $hy'/12$ 时乘以－100/96．然后在形成 y_2 时乘以23．因此包含一个“附加”（additional）误差（+0.00000004）×100 ×2 3/96 或大约 0.00000096 的“附加”误差．把这个“附加”误差在最后一位减 5 得到另一个误差(即下一个 y 的误差)为 －0.00000091．因这种现象在每一步都出现，所以在每一步这个误差大约乘以 24．

对于三阶 Adams-Bashforth 方法，多项式 ρ 和 σ 是

$$\rho(\xi)=\xi^3-\xi^2=\xi(\xi-1),$$

$$\sigma(\xi)=-\frac{1}{12}(23\xi^2-16\xi+5).$$

如果 $h\lambda=-\frac{1}{8}$，则方程 (8.17) 是

$$\xi^3-\frac{73}{96}\xi^2-\frac{16}{96}\xi+\frac{5}{96}=0.$$

它的根均小于 1[1]．于是，舍入误差在以后诸步就不引起大的误差．另一方面，如果 $h\lambda=-\frac{100}{8}$，方程 (8.17) 就变成

$$\xi^3+\frac{2204}{96}\xi^2-\frac{1600}{96}\xi+\frac{500}{96}=0.$$

这方程的一个根按绝对值将超过 $\frac{1}{3}\left(\frac{2204}{96}\right)\gg 1$，所以，很小

---

1)
$$|\xi^3|=\frac{1}{96}|73\xi^2+16\xi-5|$$
$$\leqslant\frac{1}{96}(73+16+5)\max\{|\xi^2|,1\}$$
$$=\frac{94}{96}\max\{|\xi^2|,1\}$$
$$\Rightarrow|\xi|<1.$$

的舍入误差就将迅速地被放大.

对这个问题的数值结果还不是什么特别大的值，除非选取 h 使（8.17）的解的扰动增长得没有对一般初始条件的解 $ce^{\lambda t}+t^3$ 快. 如果 $\mathrm{Re}(\lambda)\leqslant 0$，则(8.17)按绝对值没有一个根将超过 1. 反之. 如果 $\mathrm{Re}(\lambda)>0$，则主根必须十分接近于 $e^{h\lambda}$，而且没有"附加"根比主根大. 这导致我们来给实验方程 $y'=\lambda y$ 定义绝对稳定性和相对稳定性.

定义 8.4 如果(8.17)的根按绝对值 $\leqslant 1$，则称多步方法对值 $h\lambda$ 绝对稳定.

定义 8.5 如果（8.17）的"附加"(extraneous) 根按绝对值 $\leqslant$ 主根，则方法是相对稳定的.

（我们没有涉及关于 $e^{h\lambda}$ 的主根的大小，因为那是精确度的标准.）

假如再将这问题从不同的初始值求解，如 $y(0)=1$，解将是

$$y(t)=e^{\lambda t}+t^3.$$

第一个分量 $e^{\lambda t}$ 每步改变 $e^{h\lambda}$.对于 $\lambda=-1$ 以及 $h=\frac{1}{8}$，这就是 $e^{-1/8}\cong 0.88279$，而(8.17)的最大根是$\cong 0.88274$. 因此，对于这个初始值用三阶 Adams-Bashforth 方法，我们将得到较好的精确度（练习：完成这个计算). 对于 $\lambda=-100$ 以及 $h=\frac{1}{8}$，单单一步的改变量是 $e^{-100/8}\cong 0.00000044$，而我们已知(8.17)的一个根比 1 大得多. 为了精确地逼近 e^{-100t} 项，必须使用小步长，使得 $100h$ 属于这样一个区域，在该区域内 $\rho(\xi)-100h\sigma(\xi)=0$ 的一个根十分精确地逼近 e^{-100h}.于是，我们能够说明，这儿没有稳定性问题，而只有由于过大的步长 h 所引起的精确度问题. 但是，等到为了精确我们用小步长已积分到 $t=\frac{1}{4}$，这时 e^{-100t} 对十位数字是没有意义的，所以

不再需要精确地表示它．在此情况下，我们对于大的 $-h\lambda$ 值的绝对稳定的方法是注意的．

希望得到的稳定性准则．

如果我们考虑问题

$$y' = \lambda(y - F(t)) + F'(t), \tag{8.20}$$

它有解

$$y = ce^{\lambda t} + F(t).$$

我们知道，就是对于单个方程，也不能真确地说出所需要的稳定性的形式．为了保持解的一定位数的精确度，扰动不应当增长得比解还要快．如果 $F(t)$ 增长得比 $e^{\lambda t}$ 还快，又初始条件 c 不比 $F(0)$ 大，则要求 h 使得

1. 在积分 $y' = F'(t)$ 时截断误差不大；

2. (8.17) 的根使得 $|\xi_i|^n$ 的增长不比 $F(nh)$ 快．第二个要求既不是绝对稳定性准则，也不是相对稳定性准则．大多数多步方法当它们接近不稳定性时引起误差在符号上的振动．于是，在实际使用时，我们必须试验控制步长，使得这样的误差是很小的．幸而这样的误差出现使大多数步长控制程序以为有大导数发生，而把步长减小，所以，缺乏稳定性的一般准则，对问题的求解无大妨碍，但对方法的性质分类是有妨碍的．

为了得到方法的稳定性概念，我们常常考察它们的绝对稳定性区域．这种区域越大，在截断误差范围内步长 h 也可取得越大．对于一到六阶的 Adams-Bashforth 方法，绝对稳定性区域表示在图 8.1 中；对于三到六阶的 Adams-Moulton 方法，绝对稳定性区域则表示在图 8.2 中．一阶的 Adams-Moulton 方法是向后的 Euler 方法：

$$y_{n+1} = y_n + hy'_{n+1}.$$

除在圆 $|1 - h\lambda| < 1$ 内它都是稳定的．二阶 Adams-Moulton 方法是梯形法则，它在整个负半平面上是稳定的．可见隐式

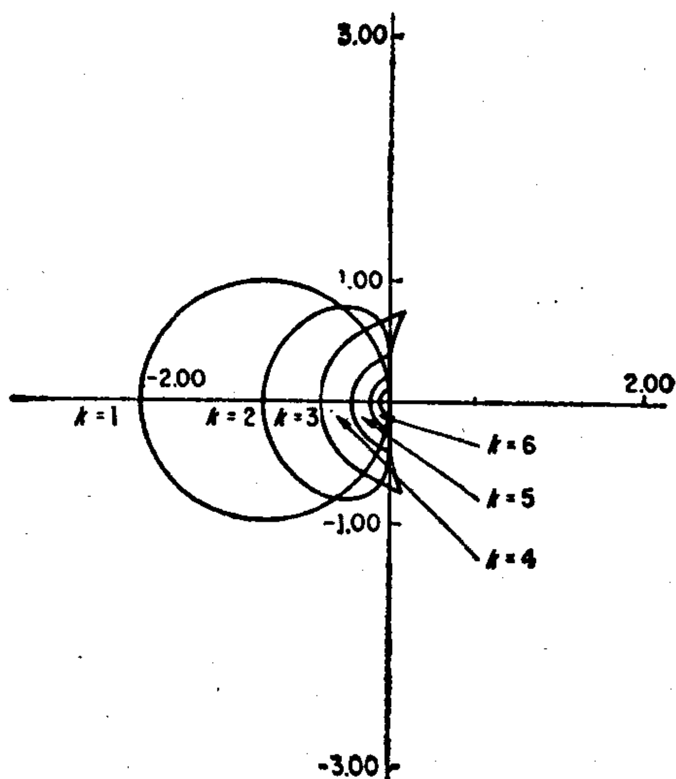


图 8.1　*k* 阶 Adams-Bashforth 方法在原点左边所指示的区域内稳定

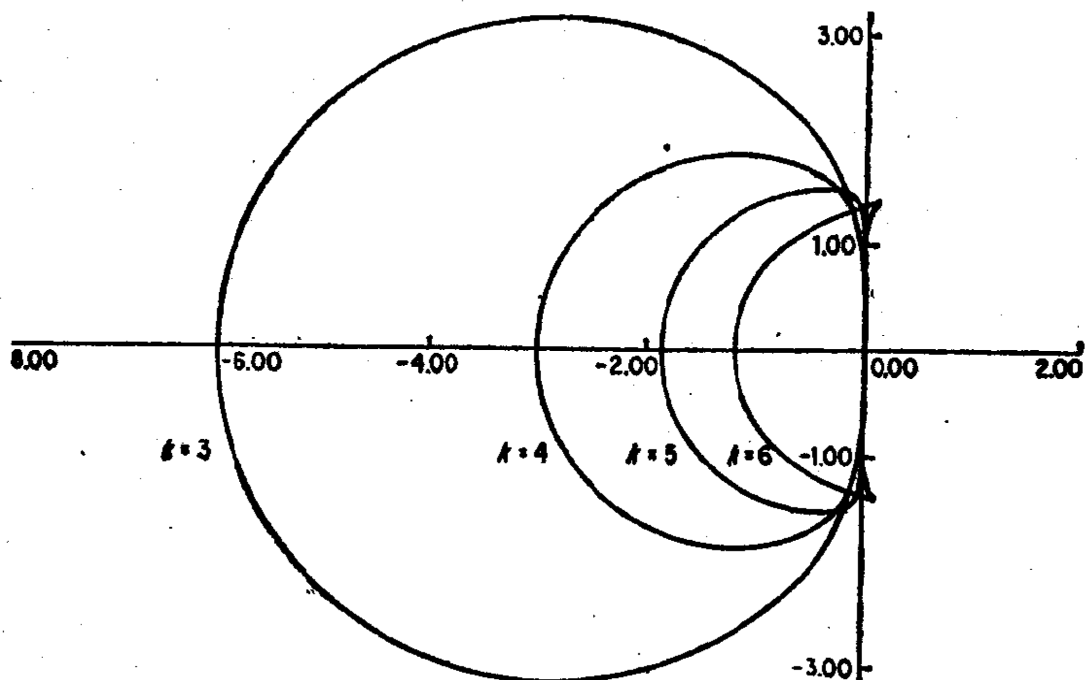


图 8.2　Adams-Moutton 方法的稳定区域。*k* 阶方法在指示的区域内是稳定的

Adams-Moulton 的方法稳定性区域比显式 Adams-Bashforth 方法的稳定性区域要大十倍或更多倍. 对于隐式方法来说，截断误差也是较小的. 这样,比显式方法大若干倍的步长,隐式方法都能用.这种步长放大通常比抵消在解校正公式时的"附加工作量"(additional effort)还要大,因为校正也许只需要两、三次函数求值.

线性方程组.

如果我们考虑方程组

$$\mathbf{y}' = A\mathbf{y}, \tag{8.21}$$

其中 A 是常矩阵,并假定可用矩阵 S 对角线化,则能把它变换为等价方程组

$$\mathbf{z}' = \Lambda\mathbf{z},$$

其中 $\mathbf{z} = S\mathbf{y}$ 且 $\Lambda = SAS^{-1}$ 是具有元素 λ_i 的对角线矩阵. 于是,(8.21) 的解是

$$\mathbf{y} = S^{-1}e^{\Lambda t}S\mathbf{y}_0,$$

其中 $e^{\Lambda t}$ 是具有元素 $e^{\lambda_i t}$ 的对角线矩阵. 如果对 (8.21) 应用多步方法,则得到扰动的向量方程

$$\sum_{i=0}^{k}(\alpha_i + hA\beta_i)\mathbf{e}_{n-i} = 0.$$

把这方程乘以 S,且记 $\mathbf{q}_n = S\mathbf{e}_n$, 我们得到

$$\sum_{i=0}^{k}(\alpha_i + h\Lambda\beta_i)\mathbf{q}_{n-i} = 0$$

这是对 $\mathbf{q}_n$ 的各个分量独立的方程组, 每一个形如 (8.16). 于是,我们在 $\mathbf{e}_n$ 的每一个分量中,将得到形式为 ξ_{ij}^n 的分量, 其中 $\xi_{ij}(j = 1, \cdots, k)$是对 $\lambda = \lambda_i$ 的(8.17)的 k 个根.因此,我们将涉及到全部 $|\xi_{ij}|$ 相对于具有最大实部的 λ_i(例如 λ_1)的主根的大小. 这是对 $h\lambda_1$ 的一个相对稳定性准则, 而不是对于其他的 $h\lambda_i$的对于大多数方法和问题,精确度要求是使得相

对稳定性不是一种限制的因素．要求精确地逼近 $e^{\lambda h}$ 是比控制“附加”根小于主根的条件更加苛刻的．如果对 $h\lambda=0$ “附加”根的一个或者更多个靠近单位圆，则这种说法多半很少是真确的．换句话说，弱稳定的或几乎弱稳定的方法多半给出相对稳定性问题，而诸根恰在单位圆内且为强稳定的方法，则不给出相对稳定性问题．

8.4. 四阶三步方法类

最一般的三步方法形如

$$\alpha_0 y_n+\alpha_1 y_{n-1}+\alpha_2 y_{n-2}+\alpha_3 y_{n-3}+h\beta_0 f_n$$
$$+h\beta_1 f_{n-1}+h\beta_2 f_{n-2}+h\beta_3 f_{n-3}=0.$$

如果要求它为四阶方法，则 α 和 β 必须满足五个方程

$$\begin{aligned}
&\alpha_0+\alpha_1+\alpha_2+\alpha_3 &&=0,\\
&3\alpha_0+2\alpha_1+\alpha_2+\beta_0+\beta_1+\beta_2+\beta_3 &&=0,\\
&9\alpha_0+4\alpha_1+\alpha_2+6\beta_0+4\beta_1+2\beta_2 &&=0,\\
&27\alpha_0+8\alpha_1+\alpha_2+27\beta_0+12\beta_1+3\beta_2 &&=0,\\
&81\alpha_0+16\alpha_1+\alpha_2+108\beta_0+32\beta_1+4\beta_2 &&=0.
\end{aligned}$$

如果还标准化使 $\sigma(1)=1$，则得到“附加”方程

$$\beta_0+\beta_1+\beta_2+\beta_3=1.$$

八个未知量的六个方程，可利用两个自由参数来求解．取 β_0 和 β_3 为自由参数．于是，我们得到

$$\alpha_0=\frac{1}{12}(-1-30\beta_0+6\beta_3),$$

$$\alpha_1=\frac{1}{12}(-9+54\beta_0+18\beta_3),$$

$$\alpha_2=\frac{1}{12}(9-18\beta_0-54\beta_3),$$

$$\alpha_3=\frac{1}{12}(1-6\beta_0+30\beta_3),$$

$$\beta_1 = \frac{1}{12}(6 + 12\beta_0 - 24\beta_3),$$

$$\beta_2 = \frac{1}{12}(6 - 24\beta_0 + 12\beta_3),$$

而误差系数为

$$c_5 = \frac{1}{120}(-1 + 10\beta_0 + 10\beta_3).$$

多项式 $\rho(\xi)$ 是

$$-\frac{1}{12}[\xi^3(1 + 30\beta_0 - 6\beta_3) + \xi^2(9 - 54\beta_0 - 18\beta_3)$$
$$- \xi(9 - 18\beta_0 - 54\beta_3) - (1 - 6\beta_0 - 30\beta_3)]$$
$$= -\frac{1}{12}(\xi - 1)[\xi^2(1 + 30\beta_0 - 6\beta_3)$$
$$+ \xi(10 - 24\beta_0 - 24\beta_3) + (1 - 6\beta_0 + 30\beta_3)].$$

二次因子在单位圆内有根的那些 β_0 和 β_3 的值是有意义的．求它最容易的办法是使边界 $|\xi| = 1$ 为一根．如果 $\beta_0 + \beta_3 = \frac{1}{6}$, $\xi = -1$ 是一个根，而 $\xi = 1$ 对 β_0 和 β_3 的任何有限值不是一个根,因为系数和是 12，与 β_0 和 β_3 是无关的．其他可能性仅仅是在单位圆上出现复数共轭对．对此，在二次因子中 ξ^2 和 1 的系数必须相等,而且判别式必定为负．
这当

$$\beta_0 = \beta_3 \text{ 以及 } 27(\beta_0 - \beta_3)^2 \leqslant 12(\beta_0 + \beta_3) - 2$$

时出现．这是一个 $\beta_0 = \beta_3$ 的线段,其中 $\beta_0 \geqslant 1/12$．用

$$1 - 6\beta_0 + 30\beta_3 = 0$$

给出一个"附加"根为零的直线,而在 $\beta_0 = \frac{3}{8}$, $\beta_3 = \frac{1}{24}$ 时，两个"附加"根均为零．这是 Adams-Moulton 方法．图 8.3 指出 (β_0, β_3) 平面内的稳定性区域,并指出常值截断误差的直线．

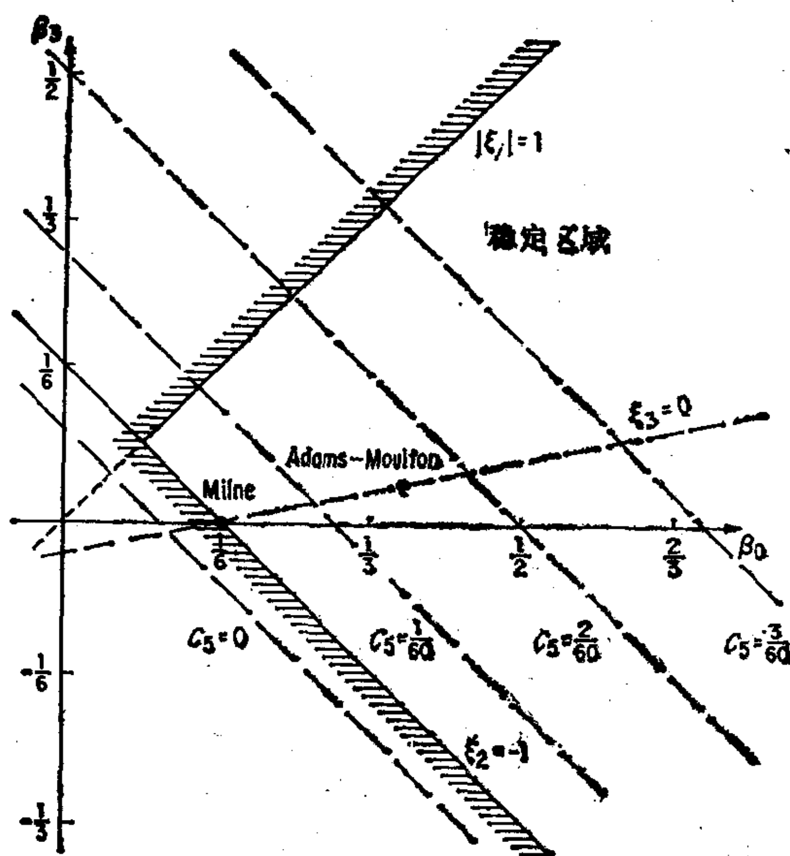


图 8.3. 对 β_0, β_3 来说，三步方法稳定

误差系数为零的直线对应于高阶方法，它是在稳定区域的外面．所以，我们断定一个稳定的三步方法不能超过四阶．如果方法是显式的，则 $\beta_0=0$．但是，这条直线也是在稳定区域的外面，所以不存在一个稳定的显式三步四阶方法．（Adams-Bashforth 三步方法只具有三阶．）

向 $\xi_2=-1$ 的直线 $\beta_0+\beta_3=\frac{1}{6}$ 移动可减小截断误差系数．但是，这使我们朝着一弱稳定的方法走．我们知道，对于 Milne 方法，第二个根按 $e^{-h\lambda/3}$ 变化．如果相信不会出现 λ，使得 $|e^{-h\lambda/3}|$ 与解的增长能够比较，则 Milne 方法是一种好的方法，因为它有最小的截断误差，又因 $\alpha_3=\beta_3=0$ 时它仅仅是

一个二步方法.但是,对于一般问题来说,因为“临界”(critical)稳定性问题，不推荐这个方法.（对通常的问题，普遍选择 Adams-Moulton 方法. 这点将在第 9 章进一步讨论.）

不希望取 β_0 和 β_3 较大的值,因为太大的值影响舍入误差也太大.

对于 k 步方法有同样的“图形”存在,所不同的是有更多个自由参数.存在一个 $k-1$ 维区域,在这个区域内 $k+1$ 阶的 k 步方法均是稳定的. 将在第 10 章证明的 Dahlquist (1956) 的主要结果指出: 如果 k 是奇数, 则在这个区域里方法的阶不能超过 $k+1$, 但是如果 k 是偶数,则存在 $[(k-2)/2]$ 维子区域,在这子区域里方法的阶可高达 $k+2$(不过不能更高). 但是, 在方法的阶为 $k+2$ 的区域里, ρ 的全部根在单位圆上, 所以, 方法仅仅是弱稳定的. $k-1$ 维区域内的任何点对于 k 步方法都可以选择, 这种选择的准则应该是为了使截断误差尽可能地小,小到与保持稳定性相适应. 可惜,这样按理想来选择通常是不可能的, 因为在问题被积出之前关于所需要的数据及其解是不知道的.

问　　题

1. 如果 $\rho(\xi)=\xi^3-\xi^2+\xi/4-\frac{1}{4}$, 求 $\sigma(\xi)$, 使得:

   (a) $\sigma(\xi)$ 为二次且方法具有三阶;

   (b) $\sigma(\xi)$ 为三次且方法具有四阶.这两个方法的误差系数是什么?

2. 如果 $\rho(\xi)=\xi^4-1$, 求次数为四的 $\sigma(\xi)$, 使得这方法具有最大阶. 求这个阶是多少? 误差系数又是什么样的?

3. 如果 $\sigma(\xi)=\xi^2$,求 $\rho(\xi)$, 使得

   (a) $\rho(\xi)$ 是二次的且阶为二;

   (b) $\rho(\xi)$ 是三次的且阶为三. 这些方法是稳定的吗?

4. 在你求出的问题 3(a) 中,方法的绝对稳定性区域是什么?

5. 确定最大阶的三步方法的系数。这方法稳定吗?

6. 利用参数 β_0 求二步三阶方法类。什么范围的 β_0 可使这些方法稳定?把误差系数表示为 β_0 的函数,画图说明 β_0 的范围、误差、Adams-Moulton 方法以及 Milne 方法。

7. 考虑 $\xi_2 = -1$ 的一般三步方法类,如图8.3所指出.证明:对于 $\beta_0 > \frac{1}{12}$, $\rho(\xi) + h\lambda\sigma(\xi) = 0$ 的一个根是 $\xi = -1 + Ah\lambda + O(h^2)$,其中 A 与 β_0 无关。A 是什么?

8. 推导如下方法的系数,给出这方法尽可能高的阶数:

$$y_{n-(1/2)} = \alpha_1 y_{n-1} + \alpha_2 y_{n-2} + \beta_1 h y'_{n-1} + \beta_2 h y'_{n-2}$$

[$y_{n-(1/2)}$ 是 $y(t_n - h/2)$ 的一个近似].

$$hy'_{n-(1/2)} = hf(y_{n-(1/2)}),$$

$$y_n = \alpha_1^* y_{n-1} + \alpha_2^* y_{n-2} + \gamma h y'_{n-(1/2)} + \beta_1^* h y'_{n-1} + \beta_2^* h y'_{n-2},$$

$$hy'_n = hf(y_n).$$

y_n 的截断误差的阶是什么?这方法稳定吗?

9. 多 值 方 法

前两章我们研究了多步方法并讨论了它们的稳定性．但是，我们仅仅对于“只用预估”(predictor only)或“只用校正”(corrector only)的方法讨论了它们的稳定性，也就是说，方法中用的显式方程或隐式方程是(在舍入误差的范围内)精确求解的．实际上，我们在显式预估公式之后用了有限次数(有时是固定的)的校正迭代．本章研究作为一般多值方法子类的这些 $P(EC)^M$ 和 $P(EC)^ME$ 方法的性质，还讨论一般多值方法的性质，而且将看到有些性质具有重要的附加好处．首先，它们给出阶不再受稳定性要求限制的多步方法的简单推广，因此，k 步方法能够是阶为 $2k$ 的方法．其次，它们指出在什么方式下不同的计算格式是等价的，这样，例如可对向后差分的Adams 方法与用前面导数值的 Adams 方法作出比较．我们将看到一种特殊格式可直接应用于高阶方程．再次，它们将提供一种研究步长改变的算法和误差估计的运算格式．

我们记得多值预估-校正方法由

$$\mathbf{y}_{n,(0)} = B\mathbf{y}_{n-1}, \tag{9.1}$$

$$\mathbf{y}_{n,(m+1)} = \mathbf{y}_{n,(m)} + \mathbf{c}G(\mathbf{y}_{n,(m)}), \qquad m \geqslant 0 \tag{9.2}$$

给出．

例．

三阶 Adams-Bashforth 预估是

$$y_{n,(0)} = y_{n-1} + \frac{h}{12}(23f_{n-1} - 16f_{n-2} + 5f_{n-3}),$$

四阶 Adams-Moulton 校正是

$$y_{n,(m+1)} = y_{n-1} + \frac{h}{24}(9f(y_{n,(m)}) + 19f_{n-1} - 5f_{n-2} + f_{n-3}).$$

于是

$$y_{n,(1)} = y_{n,(0)} + \frac{3}{8}[f(y_{n,(0)}) - (3hf_{n-1} - 3hf_{n-2} + hf_{n-3})].$$

由于 y_{n-1} 和 y_{n-2} 不需要贮存，可取 $\mathbf{y}_n = [y_n, hy'_n, hy'_{n-1}, hy'_{n-2}]^T$，那里 B 和 $\mathbf{c}$ 是

$$B = \begin{bmatrix} 1 & \frac{23}{12} & -\frac{16}{12} & \frac{5}{12} \\ 0 & 3 & -3 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \qquad \mathbf{c} = \begin{bmatrix} \frac{3}{8} \\ 1 \\ 0 \\ 0 \end{bmatrix}.$$

9.1. 误差的性态

我们考察方法的误差.

定义 9.1. 如果 $\mathbf{y}(t_n)$ 是 $\mathbf{y}_n$ 的正确值，并且计算

$$\tilde{\mathbf{y}}_{n,(0)} = B\mathbf{y}(t_{n-1}),$$
$$\tilde{\mathbf{y}}_{n,(m+1)} = \tilde{\mathbf{y}}_{n,(m)} + \mathbf{c}G(\tilde{\mathbf{y}}_{n,(m)}), \tag{9.3}$$
$$\tilde{\mathbf{y}}_n = \tilde{\mathbf{y}}_{n,(M)},$$

则局部截断误差是 $\mathbf{d}_n$，其中

$$\mathbf{d}_n = \tilde{\mathbf{y}}_n - \mathbf{y}_n(t_n).$$

注意，这就确定了用预估-校正格式求微分方程 $G(\mathbf{y}) = 0$ 的解 $\mathbf{y}(t)$ 的截断误差.

我们用

$$\mathbf{e}_n = \mathbf{y}_n - \mathbf{y}(t_n)$$

定义第 n 步的全体误差，并且定义 $\mathbf{e}_{n,(m)}$ 为 $\mathbf{y}_{n,(m)} - \tilde{\mathbf{y}}_{n,(m)}$. 从 (9.2) 减去 (9.3) 且利用中值定理得到

$$\mathbf{e}_{n,(m+1)} = \mathbf{e}_{n,(m)} + \mathbf{c}\frac{\partial G}{\partial \mathbf{y}}(\xi_m)\mathbf{e}_{n,(m)},$$

其中 ξ_m 是 $\mathbf{y}_{n,(m)}$ 和 $\tilde{\mathbf{y}}_{n,(m)}$ 中间的一点(注意 $\partial G/\partial\mathbf{y}$ 是一行向量). 于是,我们有

$$\mathbf{e}_{n,(M)} = \prod_{i=0}^{M-1}\left(I + \mathbf{c}\frac{\partial G}{\partial \mathbf{y}}(\boldsymbol{\xi}_i)\right)\mathbf{e}_{n,(0)} \tag{9.4}$$

(注意,矩阵乘法应该将具有较大下标的项放在左边).由定义我们得到

$$\mathbf{e}_{n,(0)} = \mathbf{y}_{n,(0)} - \tilde{\mathbf{y}}_{n,(0)} = B(\mathbf{y}_{n-1} - \mathbf{y}(t_{n-1})) = B\mathbf{e}_{n-1}$$

和

$$\mathbf{e}_n = \mathbf{y}_n - \mathbf{y}(t_n) = \mathbf{y}_n - \tilde{\mathbf{y}}_n + \tilde{\mathbf{y}}_n - \mathbf{y}(t_n) = \mathbf{e}_{n,(M)} + \mathbf{d}_n.$$

把最后两个方程代入 (9.4),对于 $n \geqslant 1$, 我们得到

$$\mathbf{e}_n = S_n\mathbf{e}_{n-1} + \mathbf{d}_n,$$

其中

$$S_n = \prod_{i=0}^{M-1}\left(I + \mathbf{c}\,\frac{\partial G}{\partial \mathbf{y}}(\boldsymbol{\xi}_i)\right)B. \tag{9.5}$$

由此我们看到

$$\mathbf{e}_N = \sum_{i=0}^{N}\prod_{j=i+1}^{N} S_j\mathbf{d}_i, \tag{9.6}$$

其中 $\mathbf{d}_0$ 是初值误差.

我们希望稳定性与矩阵 S_j 的"大小"有关. 例如, 若对于所有的 i 有 $\|S_j\| \leqslant 1$, 则 $\|\mathbf{e}_n\| \leqslant \sum_{i=0}^{N}\|\mathbf{d}_i\|$.

9.1.1. 预估-校正方法的稳定性

我们在第 8 章, 指出过, 稳定性与要求 $\rho(\xi)$ 的根条件成立是等价的, 而且用 (8.17) 的根定义了绝对稳定性和相对稳定性. 对于多值方法, 我们要求对方程 (9.6) 中 S_j 矩阵的特征值有类似条件. 我们考察这些条件, 并证明当校正方程被

精确求解时这些条件与前面的条件是等价的.(例如当连续迭代到收敛时.)首先，我们注意在 $f(y)=\lambda y$ 的情形，$\partial G/\partial \mathbf{y}$ 取形式

$$\left[h\frac{\partial f}{\partial y},0,\cdots,-1,0,\cdots,0\right]=h\lambda\boldsymbol{\delta}_0^T-\boldsymbol{\delta}_K^T,$$

其中 $\boldsymbol{\delta}_i$ 是第 i 个位置（由 0 开始数）为 1 而其他各处为零的列向量. $\boldsymbol{\delta}_i^T$ 为其转置. 由 (9.5) 知道 S_i 仅依赖于 $h\lambda$，所以有

$$S_j=S(h\lambda)=\left[\begin{array}{cccc:cccc}
1+h\lambda\beta_0^* & & 0 & & -\beta_0^* & & & \\
 & & 1 & & & & 0 & \\
 & 0 & & \ddots & & & & \\
 & & & 1 & & & & \\
\hdashline
h\lambda & & & & 0 & & & \\
 & & & & & 1 & & 0 \\
 & & 0 & & & & \ddots & \\
 & & & & & 0 & & 1
\end{array}\right]^{M} B$$

其中所有其他的元素均为零. 如果设

$$L_m=1+h\lambda\beta_0^*+\cdots+(h\lambda\beta_0^*)^m,$$

则有

$$S(h\lambda)=\left[\begin{array}{cccc:cccc}
L_M & & 0 & & -\beta_0^*L_{M-1} & & & \\
 & 1 & & & & & & \\
0 & & \ddots & & & & 0 & \\
 & & & 1 & & & & \\
\hdashline
h\lambda L_{M-1} & & & & -\beta_0^*h\lambda L_{M-2} & & & \\
 & & & & & & & 0 \\
 & 0 & & & & 1 & & \\
 & & & & 0 & & \ddots & 1
\end{array}\right] B$$

$$
=\left[\begin{array}{cccc:ccc}
a_1 & a_2 & \cdots & a_k & b_1 & \cdots & b_k \\
1 & & & & & & \\
 & \ddots & 0 & & & 0 & \\
 & 0 & & & & & \\
 & & 1 & 0 & & & \\
\hdashline
c_1 & c_2 & \cdots & c_k & d_1 & \cdots & d_k \\
 & & & & 1 & & \\
 & 0 & & & & \ddots & 0 \\
 & & & & 0 & & \\
 & & & & & 1 & 0
\end{array}\right]
$$

其中　$a_i = L_M\alpha_i - \beta_0^* L_{M-1}\gamma_i$,　$b_i = L_M\beta_i - \beta_0^* L_{M-1}\delta_i$,

$c_i = h\lambda(L_{M-1}\alpha_i - \beta_0^* L_{M-2}\gamma_i)$

以及　$d_i = h\lambda(L_{M-1}\beta_i - \beta_0^* L_{M-2}\delta_i)$.

方法的稳定性由这个矩阵的特征值确定.在该稳定区域内,矩阵的特征值在单位圆内或单位圆上是单重的[1]. 这样，扰动不增加,方法就是绝对稳定的. 因此,我们定义多值方法的稳定性区域为如下.

定义 9.2. 绝对稳定性区域由 $h\lambda$ 平面上那些使 $S(h\lambda)$的特征值在单位圆内或单位圆上是单重的值所组成.

定义 9.3. 相对稳定性区域由 $h\lambda$ 平面上使 $S(h\lambda)$ 的"附加"特征值都小于主特征值的那些值组成.主特征值是 $e^{h\lambda}$ 的最好近似值.

我们证明: 如果校正公式迭代到收敛，则这些特征值恰好是 $\rho^*(\xi) + h\lambda\sigma^*(\xi) = 0$ 的根.

如果校正公式迭代到收敛,则

$$L_{M-2} = L_{M-1} = \cdots = L = \frac{1}{1 - h\lambda\beta_0^*}$$

1) 我们使用"单重"这个词,意思是特征值相当于一个线性初等因子. 对独立的特征向量的情形,在单位圆上的重根不引起扰动的增加.

(这推出 $|h\lambda\beta_0^*|<1$). 在这种情形，$S(h\lambda)$ 的第 k 行是其第零行的 $h\lambda$ 倍. 如果我们作相似变换 QSQ^{-1}, 其中

$$
Q=\left[\begin{array}{ccc:ccc}
1 & & 0 & & & \\
 & \ddots & & & 0 & \\
0 & & 1 & & & \\
\hdashline
-h\lambda & & 0 & 1 & & 0 \\
 & \ddots & & & \ddots & \\
0 & & -h\lambda & 0 & & 1
\end{array}\right]
$$

则得

$$
\tilde{S}=QS(h\lambda)Q^{-1}=\left[\begin{array}{cccc:cccc}
\cdots & e_i & \cdots & & \cdots & \eta_i & \cdots & \\
1 & & 0 & & & & & \\
 & \ddots & & & & 0 & & \\
0 & & 1 & 0 & & & & \\
\hdashline
 & & & & 0 & & 0 & \\
 & 0 & & & 1 & & & \\
 & & & & & \ddots & & \\
 & & & & 0 & & 1 & 0
\end{array}\right]
$$

其中

$$
\begin{aligned}
e_i &= L\alpha_i-\beta_0^*L\gamma_i+h\lambda(L\beta_i-\beta_0^*L\delta_i)\\
&= L[\alpha_i-(\alpha_i-\alpha_i^*)+h\lambda(\beta_i-(\beta_i-\beta_i^*))]\\
&= L(\alpha_i^*+h\lambda\beta_i^*).
\end{aligned}
$$

$\tilde{S}$ 的特征值是由它的左上方子块特征值和 k 个零组成. 由于 $1/L=1-h\lambda\beta_0^*$ 且 $\alpha_0^*=-1$, 这个左上方子块矩阵是多项式

$$
-\xi^k+\sum_{i=1}^{k}\xi^{k-i}e_i=L\left[\sum_{i=1}^{k}\xi^{k-i}(\alpha_i^*+h\lambda\beta_i^*)-\frac{1}{L}\xi^k\right]
$$

$$= L[\rho^*(\xi) + h\lambda\sigma^*(\xi)]$$

的伴随矩阵[1],所以,S的特征值就是方程 $\rho^*(\xi)+h\lambda\sigma^*(\xi)=0$ 的零点. 如果做有限次校正，则校正公式的稳定性受到预估公式的影响. 但是，如果 $h\lambda=0$，第一次校正恒等于以后各次校正，稳定性由 $S=S(0)=(I-C\delta_k^T)B$ 的特征值所确定,它们是 $\xi^k\rho(\xi)=0$ 的根.

当 $h\lambda\neq 0$ 且校正迭代到不收敛时,特征值的个数比前面多项式的根多一倍.附加的 k 个根出现在 $P(EC)^M$ 方法中,因为有 $2k$ 个不同的值 $y_{n-1},\cdots,y_{n-k}$ 和 $hy'_{n-1},\cdots,hy'_{n-k}$ 一步一步贮存起来. hy'_i 的误差不直接与 y_i 的误差有关,因为在最后计算 hy'_i 的值以后 y_i 得到校正. 对每个进行计算的“独立”数出现一个根，因为每个这样的数可包含一个“独立”误差.如果 $h\lambda=0$ 或校正迭代到收敛,则有 $hy'_i=hf(y_i)$. 所以,它们的误差不是独立的(如果舍入误差忽略的话). 因此，我们期望得到 k 个根,剩下的 k 个是零,这表示: 即使由于初值或舍入引进误差，使得对于 $i<k$ 有 $hy'_i\neq hf(y_i)$，但影响在几步之后就消失.

$P(EC)^ME$ 方法相当于利用向量 $\mathbf{c}=[0,\cdots,0,1,0,\cdots]^T$ 进行附加的校正迭代. hy'_n 的值被校正，但不改变 y_n. 在这种情形，S_n 包含一个附加因子.对于方程 $y'=\lambda y$，S_n 如下形:

---

1) 伴随矩阵形如

$$A=\begin{bmatrix} a_1 & a_2 & \cdots & a_k \\ 1 & & 0 & \\ & \ddots & & \\ 0 & & 1 & 0 \end{bmatrix},$$

它的特征多项式 $\det(A-\lambda I)$ 是

$$(-1)^k(\lambda^k-a_1\lambda^{k-1}-a_2\lambda^{k-2}-\cdots-a_k).$$

$$
S_n = \left[\begin{array}{ccc|ccc}
a_1 & \cdots & a_k & b_1 & \cdots & b_k \\
1 & & & & & \\
 & \ddots & 0 & & 0 & \\
0 & & & & & \\
 & 1 & 0 & & & \\
\hline
c_1 & \cdots & c_k & d_1 & \cdots & d_k \\
 & & & 1 & & \\
 & 0 & & & \ddots & \\
 & & & & 1 & 0
\end{array}\right],
$$

其中 a_i 和 b_i 如前面的定义,而

$$c_i = h\lambda a_i, \qquad d_i = h\lambda b_j.$$

因为第 k 行是第一行的 $h\lambda$ 倍,所以,$S(h\lambda)$ 如前一样有 k 个特征根为零. 因此,我们猜想 $P(EC)^M E$ 方法比 $P(EC)^M$ 方法有更大的相对或绝对稳定性区域,因为在 $P(EC)^M E$ 方法中只需限制 k 个特征值.

当只使用有限次校正迭代时,对预估如何影响校正的稳定性问题做任何一般性的讨论都是困难的. 在文献中已列出一些利用预估来增大校正公式的稳定性区域的工作,例如,可参阅 Stetter (1968) 和 Crane 以及 Klopfenstein (1965).

9.2. 等价方法

在前一节,我们用矩阵 B 和向量 $\mathbf{c}$ 描述了多步方法. 从当前和前面几个节点得到的信息贮存在向量 $\mathbf{y}_n$ 中. 预估步的基础就是由所贮存的信息进行外插的过程(通常是多项式形式). 但是,贮存 $\mathbf{y}_n$ 分量的其他线性组合也是一样合理的,只要能够从它们得到初始的信息. 例如,在 Adams 方法中,可以贮存前面几个节点的导数值,也可以贮存它的向后差分值. 每一种信息都是另一种的线性组合.

设 T 是与 $\mathbf{y}_n$ 的维数相同的非奇异矩阵. 定义

$$\mathbf{a}_n = T\mathbf{y}_n,$$

$$\mathbf{a}_{n,(m)} = T\mathbf{y}_{n,(m)}. \tag{9.7}$$

如果将这些方程代到(9.2)和(9.1)中,则得

$$\mathbf{a}_{n,(0)} = A\mathbf{a}_{n-1}, \tag{9.8}$$

$$\mathbf{a}_{n,(m+1)} = \mathbf{a}_{n,(m)} + \mathbf{l}F(\mathbf{a}_{n,(m)}), \tag{9.9}$$

其中 $A = TBT^{-1}$, $\mathbf{l} = T\mathbf{c}$, $F(\mathbf{x}) = G(T^{-1}\mathbf{x})$.

当有误差传播方程时,我们得到

$$\mathbf{e}_n = \tilde{S}_n\mathbf{e}_{n-1} + \bar{\mathbf{d}}_n,$$

其中 $\bar{\mathbf{d}}_n$ 是变换后的方法的截断误差($=T\mathbf{d}_n$),而且

$$\begin{aligned}
\tilde{S}_n &= \prod_{i=0}^{M-1}\left(I + \mathbf{l}\frac{\partial F(\xi_i)}{\partial \mathbf{a}}\right)A \\
&= \prod_{i=0}^{M-1}\left[T\left(I + \mathbf{c}\,\frac{\partial G(T^{-1}\xi_i)}{\partial \mathbf{y}}\right)T^{-1}\right]TBT^{-1} \\
&= T\prod_{i=0}^{M-1}\left(I + \mathbf{c}\,\frac{\partial G}{\partial y}(T^{-1}\xi_i)\right)BT^{-1} \\
&= TS_nT^{-1}.
\end{aligned}$$

这样,如所期望的,“新”方法的稳定性和旧方法的稳定性是一样的. 显然,如果计算精确地进行(若没有舍入误差),则两种方法给出相同的结果. 因此,通常提出如下定义

定义 9.4. 两种多值方法经变换(9.7)到(9.9)后可以互相获得,则称这两种方法是等价的.

注意,对等价方法做变换,不影响其稳定性或截断误差,只影响舍入误差和计算量.

9.2.1. 影响表示式选取的因素

在进行数值解的逐次计算中,在实际的程序中,有许多不同的运算要执行. 明显的运算是在预估步(9.8)中用矩阵 A 的乘法,以及在每次校正步(9.9)中向量 $\mathbf{l}$ 乘纯量 F 的乘法接

着一个向量加法．实际上，只有那些为算出 y_n 和 hy'_n 所需要的 $\mathbf{a}_n$ 的分量在每次校正步中更新，其余的可以通过考察

$$\mathbf{a}_{n,(M)} = \mathbf{a}_{n,(0)} + \mathbf{l}(F(\mathbf{a}_{n,(0)}) + F(\mathbf{a}_{n,(1)}) + \cdots + F(\mathbf{a}_{n,(M-1)}))$$

来处理．假定 y_n 和 hy'_n 是 $\mathbf{a}_n$ 的两个显式分量，譬如说是在第零和第一位置，这样，只有 $\mathbf{a}_n$ 的两个分量在每次校正步中需要校正，其余的一次就能修正．

另外，一个实用方法还必须定时改变步长、阶和提供误差项的估计，使得可以选取适当的阶和步长．

步长的改变．

改变步长是一插值过程，在这过程中，对新步长所贮存的值必须在原步长所贮存的值中去找．假定使用三步方法且在点 t_n，$t_n - h$ 和 $t_n - 2h$ 上 y 及其导数值已知．如果步长减小一半，则在 t_n，$t_n - (h/2)$ 和 $(t_n - h)$ 的值可用插值法来计算．在这种情形，t_n和 $t_n - h$ 上值的计算是简单的，但是，为了近似计算 $y(t_n - (h/2))$ 和 $hy'(t_n - h/2)$，需要一个原来值的线性组合．一般用比率 α 来改变步长相当于矩阵 $C(\alpha)$ 左乘 $\mathbf{a}_n$，所以，步长改变算法如下：

$$\tilde{\mathbf{a}}_n = C(\alpha)\mathbf{a}_n, \tag{9.10}$$

其中 $\tilde{\mathbf{a}}_n$ 是新步长 αh 对 $\mathbf{a}_n$ 的调整值．

如果积分一个庞大的联立方程组，则方程 (9.10) 是比较费运算时间的．在这种情形，更经济的另一种计算格式是对可变步长用不同的积分公式．例如，我们可以计算系数 β_i，构造阶为 $k+1$ 的积分公式

$$y_n = y_{n-1} + \beta_0 h_n f_n + \beta_1 h_{n-1} f_{n-1} + \cdots + \beta_k h_{n-k} f_{n-k},$$

其中 $y_{n-i} \cong y(t_{n-i})$ 而且节点不再是等距的，因此，$h_j = t_j - t_{j-1}$．如果每 k 步改变一次步长，则对于每一步系数 β_i 一定要重复计算．但是，这与方程的个数是无关的，所以，若有一个庞大的方程组，则它比用 (9.10) 更经济，因 (9.10) 必须用于

每个因变量.

应当注意,这两种方法是不等价的.在第一种情形,$\mathbf{a}_n$ 的所有原来值在用插值法时也许全部用来形成新值 $\tilde{\mathbf{a}}_n$. 因此,在最远的节点上的值可能影响许多贮存值 $\tilde{\mathbf{a}}_n$. 在第二种情形,最远的值在下一个步长以后被丢掉,而且不再有影响.

我们用下面的例子来说明这一点. 三步 Adams-Bashforth 公式由

$$B = \begin{bmatrix} 1 & \frac{23}{12} & -\frac{16}{12} & \frac{5}{12} \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

给出. 如果步长减半,则插值可用

$$C\left(\frac{1}{2}\right) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{3}{16} & \frac{3}{8} & -\frac{1}{16} \\ 0 & 0 & \frac{1}{2} & 0 \end{bmatrix}$$

左乘 $\mathbf{y}_n$ 来实现. 第三行利用了

$$\tilde{h}y(t-\tilde{h}) = \frac{3}{16}hy'(t) + \frac{3}{8}hy'(t-h)$$

$$-\frac{1}{16}hy'(t-2h) + O(h^4),$$

其中 $\tilde{h} = h/2$. 假定用步长 h 得到 $\mathbf{y}_n$ 的值,而且积分方程 $y' = 0$, 首先把步长减半, 然后完成两个积分步. 其结果是以矩阵 $BBC\left(\frac{1}{2}\right)$ 乘 $\mathbf{y}_n$, 这儿 $BBC\left(\frac{1}{2}\right)$ 是

$$BBC\left(\frac{1}{2}\right)=\begin{bmatrix}1 & \frac{23}{192} & -\frac{26}{192} & \frac{11}{192}\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & \frac{1}{2} & 0 & 0\end{bmatrix}.$$

第一行告诉我们初值误差怎样影响结果.

另外,我们可使用近似公式

$$y\left(t+\frac{h}{2}\right)=y(t)+\frac{h}{24}(17y'(t)-7y'(t-h)+2y'(t-2h))+O(h^4)$$

和

$$y\left(t+\frac{h}{2}\right)=y(t)+\frac{h}{72}\left(64y'(t)-33y'\left(t-\frac{h}{2}\right)+5y'\left(t-\frac{3h}{2}\right)\right)+O(h^4).$$

利用图 9.1 中标明的点 $n-3$, $n-2$ 和 $n-1$ 来得到 n,又利用 $n-2$, $n-1$ 及 n 来得到 $n+1$.

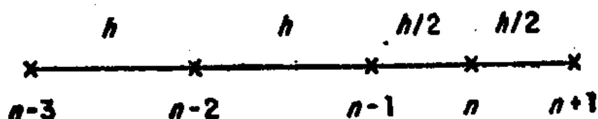


图 9.1. 可变公式的步长改变

第一步相当于使用矩阵

$$B_1=\begin{bmatrix}1 & \frac{17}{24} & -\frac{7}{24} & \frac{2}{24}\\ 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}$$

而第二步相当于使用矩阵

$$B_2 = \begin{bmatrix} 1 & \frac{64}{72} & -\frac{33}{72} & \frac{5}{72} \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \end{bmatrix}.$$

（在第二步，将 hy' 的贮存值减小一半来得到 $\tilde{h}y'$.）

这两步的结果与将矩阵 B_2B_1 应用到初始值是等价的．这一次的结果是

$$B_2B_1 = \begin{bmatrix} 1 & \frac{9}{36} & -\frac{8}{36} & \frac{3}{36} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \end{bmatrix}$$

我们看到 $B_2B_1 \neq BBC\left(\frac{1}{2}\right)$，所以两种步长减半的方法是不等价的．

这种不同的结果在整个稳定性性质上也可能是不同的，但是直到目前为止，还没有研究具有变步长的多步方法的稳定性．

阶的改变．

方法的阶是由它的系数确定的．一般高阶方法需要大量的前面的点．充分大的一组前面的信息总是可以贮存的．于是 5 × 5 阶矩阵和 5 维向量

$$B = \begin{bmatrix} 1 & \frac{23}{12} & -\frac{16}{12} & \frac{5}{12} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad \mathbf{c} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

刚好是三步 Adams-Bashforth 方法的表示形式，虽然它还要贮存 $\mathbf{y}_n$ 中 hy'_{n-3} 的值．实际上，如果表达式中所用的信息开始时已计算好，就不费什么事情，只要耗费一点贮存空间来贮存已经计算好的信息．有些表示形式需要在每一步对贮存的信息进行处理．在那种情形，除恰好要提高阶的前几步以外，逐步保留不必要的数据是不妥当的．因此，如果使用向后差分的 Adams 方法，则只有那些所需要的差分是应该保留的．

估计误差．

我们已经知道，局部截断误差是与解 $y(t)$ 的导数成比例的．这导数可以用数值微分公式来估计．［见 Hildebrand (1956)，第 3.3 节．］它是贮存的信息 $\mathbf{a}_n$ 的一个线性组合．可以期望估计几种不同阶方法中的误差，这样就必须估计若干个不同的导数．如果要估计 q 个不同的导数，我们用运算 $D\mathbf{a}_n$ 来表示，其中 D 是 $q \times (\mathbf{a}_n)$ 的维数阶的矩阵．有些 D 的表示形式是特别简单的．例如，如果 hy' 的向后差分保留在 $\mathbf{a}_n$ 中，则差分 $\nabla^j hy'$ 是 $h^{j+1}y^{(j+1)}$ 的一种估计，所以，D 的每一行仅需含有一个非零分量．

9.2.2. Adams 方法的向后差分表示式

四阶的三步 Adams-Bashforth-Moulton 方法的 B 和 $\mathbf{c}$ 矩阵已知为

$$B = \begin{bmatrix} 1 & \frac{23}{12} & -\frac{16}{12} & \frac{5}{12} \\ 0 & 3 & -3 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \qquad \mathbf{c} = \begin{bmatrix} \frac{3}{8} \\ 1 \\ 0 \\ 0 \end{bmatrix}.$$

如果使用三阶 Adams-Moulton 校正公式，则读者可以证明：B 不改变，而 $\mathbf{c}$ 则变为 $[5/12, 1, 0, 0]^T$．如果变换为向后差分，则有

$$\begin{bmatrix} y_n \\ hy'_n \\ \nabla hy'_n \\ \nabla^2 hy'_n \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} y_n \\ hy'_n \\ hy'_{n-1} \\ hy'_{n-2} \end{bmatrix} = T\mathbf{y}_n,$$

而新的方法由

$$A = TBT^{-1} = \begin{bmatrix} 1 & 1 & \frac{1}{2} & \frac{5}{12} \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

至少还有

$$\mathbf{l} = TC = \begin{bmatrix} \beta_0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \tag{9.11}$$

给出，其中对四阶校正公式有 $\beta_0 = \frac{3}{8}$，对三阶校正公式有 $\beta_0 = \frac{5}{12}$. 运算次数不像以前那么多. A 乘 $\mathbf{a}_n$ 的乘法只包含两次乘法、五次加法和四次贮存，因为 A 的最后三行能用如下技巧来处理:

$$a_{n-1,3} + a_{n-1,2} \rightarrow a_{n,2},$$

$$a_{n,2} + a_{n-1,1} \rightarrow a_{n,1},$$

其中 $a_{n-1,i}$ 是 $\mathbf{a}_{n-1}$ 的第 i 个分量.

这种表示法比用导数值的表示法每步需要的运算较少. 它也是估计导数的一种最方便的表示，因为向后差分直接提供了这种估计. 可惜对于变步长来说，这种表示形式并不方便，因为用插值求得对新步长的向后差分的矩阵 $C(\alpha)$ 是琐碎的. 除前两行和前两列的非零元素在对角线上之外，矩阵

$C(\alpha)$ 其余部分的上三角形全是非零元素.对于上面 4×4 阶的情况,

$$C(\alpha)=\begin{bmatrix}1&0&0&0\\0&\alpha&0&0\\0&0&\alpha^2&\dfrac{\alpha^2(1-\alpha)}{2}\\0&0&0&\alpha^3\end{bmatrix}.$$

第三行是根据

$$\tilde{h}\tilde{\nabla}y'=\alpha^2h\nabla y'+\frac{\alpha^2(1-\alpha)h\nabla^2y'}{2}+O(h^4)$$

得到的,其中 $\tilde{h}=\alpha h$, ∇ 是基于步长 h 的差分算子,而 $\tilde{\nabla}$ 是基于步长 $\tilde{h}$ 的.

向后差分表示的一般 Adams 方法.

这种表示的 Adams 方法的一般形式由

$$A=\begin{bmatrix}1&\gamma_0&\gamma_1&\gamma_{k-2}\\&1&1&1\\&&1&\\&&&\ddots\\&&&1\end{bmatrix},\quad \mathbf{l}=\begin{bmatrix}\gamma_q\\1\\1\\\vdots\\1\end{bmatrix}$$

给出,其中 $q=k-2$ 或 $k-1$, 这取决于校正公式的阶,是与预估公式的阶一样为 $(k-1)$, 还更高一阶为 (k).

这种表示的优点是 $\mathbf{a}_n$ 的分量有变得更小的可能, 因为 hf_n 的第 j 个差分是 $h^{j+1}f_n^{(j+1)}+O(h^{j+2})$. 因此, 在浮点运算中,与 $\mathbf{a}_n$ 的分量的大小成比例, 舍入误差也只在前几个分量中有意义. 所以,如果一定要用多倍长精度,那么, $\mathbf{a}_n$ 的最后几个分量可以不必要求高于单精度.

应用均差作为简化改变步长算法的一种方法,已由 Krogh (1969) 提出 [见 Hildebrand (1956), 第 2 章]. 它在每一步对

于每个因变量引进了 k 个乘法，不过对于庞大的方程组来说也许是可取的，因为其中计算预估-校正系数的总的计算量是少的．

9.2.3. Adams 方法的 Nordsieck 形式

Nordsieck（1962）提出贮存 y_n，y'_n，$hy''_n/2$，$\cdots$，$h^{k-2}\times y_n^{(k-1)}/(k-1)!$ 的近似值代替贮存因变量及其导数值．他的出发点是使步长改变简单．Adams 方法和 Nordsieck 方法之间的对应关系可用如下变换来说明．

在 $(k-1)$ 步 Adams 方法中用到的一组贮存 y_n，hy'_n，$\cdots$，hy'_{n-k+2} 的值唯一确定一个与这些 y 和 y' 符合的 $(k-1)$ 次多项式．这个多项式同样可用它在 t_n 的值及前 $(k-1)$ 个导数值来表示．于是，对于 $k-1$ 次多项式有变换

$$\mathbf{a}_n = T\mathbf{y}_n,$$

其中 $\mathbf{y}_n=[y_n, hy'_n, \cdots, hy'_{n-k+2}]^T$，$\mathbf{a}_n=[y_n, hy'_n, \cdots, h^{k-1}y_n^{(k-1)}/(k-1)!]^T$．对于由（9.11）给出的三阶预估，四阶校正方法，我们有

$$T=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & \frac{3}{4} & -1 & \frac{1}{4} \\ 0 & \frac{1}{6} & -\frac{1}{3} & \frac{1}{6} \end{bmatrix}$$

和

$$T^{-1}=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & -2 & 3 \\ 0 & 1 & -4 & 12 \end{bmatrix}.$$

由此得到

$$A = TBT^{-1} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ & 1 & 2 & 3 \\ 0 & & 1 & 3 \\ & & & 1 \end{bmatrix}, \mathbf{l} = T\mathbf{c} = \begin{bmatrix} \frac{3}{8} \\ 1 \\ \frac{3}{4} \\ \frac{1}{6} \end{bmatrix}. \tag{9.12}$$

这好像为了简化步长变化过程而增加用矩阵 $\tilde{A}$ 相乘的工作量．其实，并不像表现得那样坏，因左乘 A 只含有加法．A 是 Pascal 三角形矩阵，它的 (i, j) 元素是 $\binom{j}{i}$ $(k>j \geqslant i \geqslant 0)$．这表明 $h^i y_n^{(i)}/i!$ 的 $(k-1)$ 阶预估公式用 Taylor 级数给出为

$$\frac{h^i y_n^{(i)}}{i!} = \sum_{j=i}^{k-1} \frac{\binom{j}{i} h^j y_{n-1}^{(j)}}{j!} + O(h^k).$$

鉴于关系式

$$\binom{j}{i} + \binom{j}{i+1} = \binom{j+1}{i+1},$$

我们可用如下步骤来计算 $A\mathbf{a}$:

$$a_{k-1} + a_{k-2} \to a_{k-2}$$
$$\cdots$$
$$\underline{a_1 + a_0 \to a_0}$$
$$a_{k-1} + a_{k-2} \to a_{k-2}$$
$$\cdots$$
$$\underline{a_2 + a_1 \to a_1}$$
$$a_{k-1} + a_{k-2} \to a_{k-2}$$
$$\cdots$$

$$\frac{a_3 + a_2 \to a_2}{\cdots}$$

$$\underline{a_{k-1} + a_{k-2} \to a_{k-2}}$$

其中 a_i 是 $\mathbf{a}$ 的第 i 个元素．这需要 $k(k-1)/2$ 次加法和贮存．它比向后差分方法多，因为对于同样的步骤向后差分只需要 $2k-3$ 次加法和 $k-1$ 次贮存．

改变步长 α 倍的矩阵 $C(\alpha)$ 是

$$C(\alpha) = \begin{bmatrix} 1 & & & & \\ & \alpha & & 0 & \\ & & \alpha^2 & & \\ & 0 & & \ddots & \\ & & & & \alpha^{k-1} \end{bmatrix}$$

这方法还有向后差分方法的舍入误差小及可得到导数的直接估计的其它一些优点．还将看到，它对于 9.2.5 中将要讨论的高阶方程是有用的．

9.2.4. 改进的多步方法

前面提到的 Dahlquist 定理指出：强稳定的 k 步方法不能超过 $k+1$ 阶．k 步方法发展起来的表示法包含一些不严格的多步方法．方程 (9.1) 中的 $\mathbf{c}$ 包含由方法 (β_0^*) 确定的一个常数、一个 1 和 $2k-2$ 个等于零的元素．如果让 $\mathbf{c}$ 的其它元素不为零，那么会发生什么情况呢？

通过一个例子能够很好地看到这一点．我们从 (9.11) 给出的四阶三步 Adams-Bashforth-Moulton 格式开始，不用 $[y_n, hy'_n, hy'_{n-1}, hy'_{n-2}]^T$ 表示三次多项式，而用 $\mathbf{a}_n = [y_n, hy'_n, y_{n-1}, hy'_{n-1}]^T$，其变换为

$$T=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 1 & -\frac{5}{12} & -\frac{8}{12} & \frac{1}{12}\\ 0 & 0 & 1 & 0\end{bmatrix}.$$

于是方法由

$$A=TBT^{-1}=\begin{bmatrix}-4 & 4 & 5 & 2\\ -12 & 8 & 12 & 5\\ 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\end{bmatrix},\quad l=T\mathbf{c}=\begin{bmatrix}\frac{3}{8}\\ 1\\ -\frac{1}{24}\\ 0\end{bmatrix}$$

给出，这相当于由

$$y_{n,(0)}=-4y_{n-1}+5\bar{y}_{n-2}+4hf_{n-1}+2hf_{n-2},$$

$$D_{(0)}=hf(y_{n,(0)},t_n)-(-12y_{n-1}+12\bar{y}_{n-2}$$
$$+8hf_{n-1}+5hf_{n-2}),$$

$$y_{n,(1)}=y_{n,(0)}+\frac{3D_{(0)}}{8},$$

$$\cdots$$

$$D_{(m)}=hf(y_{n,(m)},t_n)-hf(y_{n,(m-1)},t_n),$$

$$y_{n,(m+1)}=y_{n,(m)}+\frac{3D_{(m)}}{8},$$

$$\cdots$$

$$y_n=y_{n,(M)},$$

$$f_n=f(y_{n,(M-1)}),$$

$$\bar{y}_{n-1}=y_{n-1}-\frac{D_{(0)}+D_{(1)}+\cdots+D_{(M-1)}}{24}$$

给出的改进的多步方法．在最后一步之前，即对前面的一个函数校正之前，它与通常的多步方法是完全等同的．正是由于这一步使方法变得稳定．由于这个方法与三步的 Adams-Bashforth-Moulton 预估-校正格式等价，我们知道，这方法是稳

定的，而且是四阶的，虽然它只是一个二步方法.

我们从阶为 $2k$ 的 $(2k-1)$ 步 Adams-Bashforth-Moulton 方法开始，把这方法变换成为相同阶的 k 步改进的多步方法. 于是，只要放宽多步方法的定义，考虑到 $\mathbf{c}$ 中有附加的非零分量， 那么稳定的阶为 $2k$ 的 K 步方法是可能有的. 正是由于这个原因，我们提出多值方法的名字[1]. 方法的重要特征是贮存一步的值去计算下步的值而不是贮存几步的值. 一阶方程的 k 值方法可以是 k 阶的.

9.2.5. 高阶方程

像用单步方法一样，把高阶方程归结成一阶方程组之后可以应用多值方法. 它们也可以直接应用于高阶方程.

我们考虑已有的多值方法的计算公式. 在向量 $\mathbf{a}_{n-1}$ 中含有解的先前性态的信息，它可以看成为在 t_{n-1} 的邻域内对解进行近似的一个次数为 $k-1$ 的多项式的表示.（假定在 $\mathbf{a}_{n-1}$ 中有 k 个分量）预估步 $\mathbf{a}_{n,(0)} = A\mathbf{a}_{n-1}$ 是用这个多项式（或者是它的其它近似）对于 t_n 的 $\mathbf{a}$ 值的外插. 如果使用可能的最高阶为 $(k-1)$ 的预估公式，则 $\mathbf{a}_n$ 将表示同样的多项式. 如果微分方程的解是次数为 $k-1$ 或次数更小的多项式，且 $\mathbf{a}_{n-1}$ 正确地表示了那个解，则除舍入误差外，$\mathbf{a}_n$ 也表示那个解. 如果微分方程的解不是一个次数 $\leqslant k-1$ 的多项式，那么就有大小为 $O(h^k)$ 的截断误差. 显然，用它自己本身作预估不能是稳定的，因为它完全没有考虑到微分方程. 因此，通过量测由 $\mathbf{a}_{n,(0)}$ 给出的近似多项式是否满足 t_n 的方程，从而对前面的步传播来的和由舍入及截断引进的误差进行校正. 量测由函数 $F(\mathbf{a}_{n,(0)})$ 来完成，如果满足微分方程，则 $F(\mathbf{a}_{n,(0)})$

1）这个术语是由 Berkeleyin B. Parleff 教授所提出的

为零．当 F 不为零时，由校正方程 $\mathbf{a}_{n,(m+1)}=\mathbf{a}_{n,(m)}+\mathbf{l}F(\mathbf{a}_{n,(m)})$ 把 $F(\mathbf{a}_{n,(0)})$ 的倍数加到 $\mathbf{a}_{n,(0)}$ 上．如果这个过程收敛，则收敛到 $\mathbf{a}_n$，使得 $F(\mathbf{a}_n)=0$．这个过程的稳定性取决于矩阵

$$S_n=\sum_{i=0}^{M-1}\left(I+\mathbf{l}\frac{\partial F}{\partial \mathbf{a}}(\xi_i)\right)A,$$

而且已知对一阶方程，存在 $\mathbf{l}$，使得这个过程是稳定的，因为这是由稳定的多步方法导出的．

我们自然要问，如果把完全一样的方法用于高阶方程，会发生什么问题．假定有微分方程

$$y^{(p)}=f(y,y',y'',\cdots,y^{(p-1)},t).$$

我们使用一种表示法，其中 $\mathbf{a}$ 的前面 $p+1$ 个分量为 y，hy'，$h^2y''/2$，$\cdots$，$h^py^{(p)}/p!$．定义

$$F(\mathbf{a})=\frac{h^p}{p!}f\left(a_0,\frac{a_1}{h},\frac{2a_2}{h^2},\cdots,\frac{(p-1)!a_{p-1}}{h^{p-1}},t\right)-a_p,$$

其中 a_i 是 $\mathbf{a}$ 的第 i 个分量，i 从 0 数起．我们可以使用同样的预估过程，即

$$\mathbf{a}_{n,(0)}=A\mathbf{a}_{n-1}$$

以及由

$$\mathbf{a}_{n,(m+1)}=\mathbf{a}_{n,(m)}+\mathbf{l}F(\mathbf{a}_{n,(m)})$$

给出的类似的校正过程，其中 $F(\mathbf{a})$ 用来表示微分方程不被 $\mathbf{a}$ 局部满足的数量，且 $\mathbf{l}$ 是为了达到稳定性和精确度而选择的．如果再研究误差传播，则得方程

$$\mathbf{e}_n=S_n\mathbf{e}_{n-1}+\mathbf{d}_n,$$

其中 $\mathbf{d}_n$ 是局部截断误差，S_n 由

$$S_n=\prod_{i=1}^{M}\left(I+\mathbf{l}\frac{\partial F}{\partial \mathbf{a}}(\xi_i)\right)A$$

给出．因为

$$\frac{\partial F}{\partial \mathbf{a}}=-\boldsymbol{\delta}_p^T+\sum_{q=0}^{p-1}\frac{\partial f}{\partial y^{(q)}}\boldsymbol{\delta}_q^T h^{p-q}\frac{q!}{p!},$$

所以我们得到

$$S_n = S + O(h),$$

其中

$$S = (I - \mathbf{l}\delta_p^T)^M A.$$

方法的稳定性取决于 S 的特征值. 如果 $\mathbf{d}_n$ 不比 $O(h^{p+1})$ 坏,又 S 的全部特征值都在单位圆内或单位圆上,除了 1 是 p 重特征值外,都是单重根,则这方法当 $h \to 0$ 时收敛.第 10 章将证明如下结果:

如果 A 使得"预估"过程为最大的可能阶是 $(k-1)$,则可选取 $\mathbf{l}$,使得

$$S = (I - \mathbf{l}\delta_p^T)^M A$$

的 $k-p$ 个特征值取任何选取的值. S 的其他 p 个特征值为 1. 这 $k-p$ 个特征值唯一确定 $\mathbf{l}$ 的最后 $k-p$ 个分量,而 $\mathbf{l}$ 的前面 p 个分量可选择得使截断误差为 $O(h^{k+1})$. 如果 S 的 $k-p$ 个"附加"特征值在单位圆内或单位圆上为单根,对 p 阶方程这样的方法将收敛,且全体误差为 $O(h^{k+1-p})$.

(如果 f 缺少某些导数,又如果初始值的选取很有经验,则方法的阶可稍微增加.)

于是看到,不管使用什么表示形式,我们总能选取"附加"特征值具有任何值. 最通常的选取是使其为零,在这种情形,Adams 方法类似的阶更高.

S 对 p 阶方程有 p 个等于 1 的主特征值. 如果对于一般的线性 p 阶微分方程

$$y^{(p)} + a_1 y^{(p-1)} + \cdots + a_p y = 0,$$

已得到稳定性矩阵,则 p 个主根将接近 $e^{h\lambda i}(i = 1, \cdots, p)$,其中 λi 是

$$\lambda^p + a_1\lambda^{p-1} + \cdots + a_p\lambda^0 = 0$$

的根. 至少到 p 阶导数是显式出现的方法,其表示式是令人

满意的，只要当计算 F 时避免为了得到所需要的导数在每一步进行线性变换．对使用 $\mathbf{a}=[y, hy', \cdots, h^{k-1}y^{(k-1)}/(k-1)!]^T$ 的表示式的 $p=1, 2, 3$ 和 4 的 $\mathbf{l}$ 值，在下面的表 9.1 中给出．全部“附加”根为零，所以对于 $p=1$，这些方法与 Adams 方法是等价的．局部截断误差为 $O(h^{k+q})$，其中 q 是使微分方程可记为 $y^{(p)}=f(y, y', \cdots, y^{(p-q)}, t)$ 的最大整数（与 q 有关的内容，将在第10章说明）．全体误差是 $O(h^{k+q-p})$．

表 9.1　对于标准形式的多值方法的系数

p	k	l_0	l_1	l_2	l_3	l_4	l_5	l_6	l_7
1	3	$\frac{5}{12}$	1	$\frac{1}{2}$					
	4	$\frac{3}{8}$	1	$\frac{3}{4}$	$\frac{1}{6}$				
	5	$\frac{251}{720}$	1	$\frac{11}{12}$	$\frac{1}{3}$	$\frac{1}{24}$			
	6	$\frac{95}{288}$	1	$\frac{25}{24}$	$\frac{35}{72}$	$\frac{5}{48}$	$\frac{1}{120}$		
	7	$\frac{19087}{60480}$	1	$\frac{137}{120}$	$\frac{5}{8}$	$\frac{17}{96}$	$\frac{1}{40}$	$\frac{1}{720}$	
	8	$\frac{5257}{17280}$	1	$\frac{49}{40}$	$\frac{203}{270}$	47	$\frac{7}{144}$	$\frac{7}{1440}$	$\frac{1}{5040}$
2	4	$\frac{1}{6}$	$\frac{5}{6}$	1	$\frac{1}{3}$				
	5	$\frac{19}{120}$	$\frac{3}{4}$	1	$\frac{1}{2}$	$\frac{1}{12}$			
	6	$\frac{3}{20}$	$\frac{251}{360}$	1	$\frac{11}{18}$	$\frac{1}{6}$	$\frac{1}{60}$		
	7	$\frac{863}{6048}$	$\frac{665}{1008}$	1	$\frac{25}{36}$	$\frac{35}{144}$	$\frac{1}{24}$	$\frac{1}{360}$	
	8	$\frac{1925}{14112}$	$\frac{19087}{30240}$	1	$\frac{137}{180}$	$\frac{5}{16}$	$\frac{17}{240}$	$\frac{1}{120}$	$\frac{1}{2520}$
3	5	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{5}{4}$	1	$\frac{1}{4}$			
	6	$\frac{3}{80}$	$\frac{19}{40}$	$\frac{9}{8}$	1	$\frac{3}{8}$	$\frac{1}{20}$		
	7	$\frac{221}{5040}$	$\frac{9}{20}$	$\frac{251}{240}$	1	$\frac{11}{24}$	$\frac{1}{10}$	$\frac{1}{120}$	
	8	$\frac{2185}{46368}$	$\frac{863}{2016}$	$\frac{95}{96}$	1	$\frac{25}{48}$	$\frac{49}{336}$	$\frac{1}{48}$	$\frac{1}{840}$
4	6	$\frac{1}{30}$	$\frac{1}{10}$	1	$\frac{5}{3}$	1	$\frac{1}{5}$		
	7	$\frac{16}{630}$	$\frac{3}{20}$	$\frac{19}{20}$	$\frac{3}{2}$	1	$\frac{3}{10}$	$\frac{1}{30}$	
	8	$\frac{11}{630}$	$\frac{221}{1260}$	$\frac{9}{10}$	$\frac{251}{180}$	1	$\frac{11}{30}$	$\frac{1}{15}$	$\frac{1}{210}$

例．

考虑由表 9.1 得到的二阶方程的四值方法．它贮存 y_n，hy'_n，$h^2y''_n/2$，和 $h^3y'''/6$ 的值．向量 $\mathbf{l}$ 是 $\left[\frac{1}{6}, \frac{5}{6}, 1, \frac{1}{3}\right]^T$．如

果用

$$\begin{bmatrix} y_n \\ y_{n-1} \\ \frac{h^2y_n''}{2} \\ \frac{h^2y_{n-1}''}{2} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & -3 \end{bmatrix} \begin{bmatrix} y_n \\ hy_n' \\ \frac{h^2y_n''}{2} \\ \frac{h^3y_n'''}{6} \end{bmatrix}$$

或 $\mathbf{y}_n = Q\mathbf{a}_n$ 变换到贮存 $y_n, y_{n-1}, h^2y_n''/2$ 和 h^2y_{n-1}'' 的表示式，则得

$$B = QAQ^{-1} = \begin{bmatrix} 2 & -1 & 2 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

$$\mathbf{c} = Q\mathbf{l} = [1/6, 0, 1, 0]^T.$$

这与用“预估”方程

$$y_{n,(0)} = 2y_{n-1} - y_{n-2} + h^2y_{n-1}''$$

是等价的。这是对特殊的二阶方程 $y''=f(y,t)$ 的一种 Stormer (1907 和 1921) 显式公式. (在这种表示形式中利用比例导数，我们可把它用于 y' 出现的一般二阶方程)校正公式与

$$y_{n,(m+1)} = 2y_{n-1} - y_{n-2} + \frac{h^2}{12}[f(y_{n,(m)}) + 10y_{n-1}'' + y_{n-2}'']$$

等价。这是对特殊二阶方程的一种 Cowell 隐式方法 [Cowell 和 Crommelin (1910)]. 对于特殊的二阶方程的这些方法的一般形式在 Henrici (1962) 的第六章讨论了. 我们将利用这种比例导数的表示形式在第 10 章讨论这些方法的理论.

9.3. 自动控制步长和阶

一个实现多值方法的程序必须使用开始值计算、变步长和变阶作为必要的技术. 本节讨论这些技术并且通过对一阶

方程的一个通用自动程序说明之.

用哪类等价方法的选择取决于该问题的本身，在编制通用的程序时，关于有待积分的问题的性态，几乎一无所知，所以，"附加"特征值为零的 Adams 方法通常是最好的选择.(对于特殊问题的其他方法，在第 11 章讨论.)但是，本节要说的东西以及后面给出的程序，如果适当改变常数，同样适用于其他方法.

如果"预估"公式的阶加上校正迭代的次数超过"校正"公式的阶，则 q 阶校正的局部截断误差是 $c_{q+1}h^{q+1}y^{(q+1)}+O(h^{q+2})$.对于 Adams-Moulton 方法 $c_{q+1}=r_q^*$(见表 7.4).为了估计 $y^{(q+1)}$，至少需要 $q+2$ 个值.如果在 $\mathbf{a}_{n-1}$ 中拥有 $q+1$ 个值，那么推导 $\mathbf{a}_n$ 时所产生的一个附加值足够用来估计局部截断误差.由于需要比较几种不同阶的误差，我们使用向后差分或 $\mathbf{a}$ 中的比例导数.因为用比例导数步长改变是比较简单的，故对于 q 阶校正，选用 $\mathbf{a}=[y, hy', \cdots, h^q y^{(q)}/q!]^T$.对 $\mathbf{a}$ 的 $q+1$ 个值来说，"预估"公式也可以是 q 阶的，所以，即使只用一次校正迭代，误差是 $c_{q+1}h^{q+1}y^{(q+1)}+O(h^{q+2})$.这些方法的 $\mathbf{l}$ 值可从表 9.1 得到，其中 $l_1, \cdots, l_q$ 是从 $K=q$ 行得到的，而 l_0 是从 $K=q-1$ 行得到的(表 9.1 给出方法的系数，其中校正公式比预估公式高一阶).

用这种表示形式，在每一步中，$\mathbf{a}$ 的最后系数的改变量 ∇a_q 是关于 $h^{q+1}y^{(q+1)}/q!$[1] 的一种估计.因此，若控制单步截断误差小于 ε，必须选取 h，使得

$$c_{q+1}q!\nabla a_q \leqslant \varepsilon,$$

其中 ∇a_q 是 $\mathbf{a}$ 的最后分量的向后差分.当积分一个方程组

1) 要证明这点，必须证明 y_n 的数值解是由 $y(t_n)+h^S\delta(t_n, h)$ 给出的，其中 $\delta(t_n,h)$ 对 t 至少有 $q+2-S$ 次连续导数.虽然除特殊情况外不能证明，但方法似乎是成功的.

时，希望分别控制每个分量的误差，所以，使

$$c_{q+1}q!\left\|\frac{\nabla a_q}{\omega}\right\|_2 \leqslant \varepsilon,$$

其中方程组的每个元有一个 ∇a_q 分量和权分量 ω. $\|\cdot\|_2$ 是 L_2-模. 使用 L_2-模是因为在使用计算机上来计算稍微快些. 如果最大模计算比较快，它完全可以同等使用. 注意，如果使用 L_2-模，仅需算出$(\|\cdot\|_2)^2$ 来作试验 (9.13).

基本的步长控制程序是积分一步并完成试验 (9.13). 如果试验成功，则采用这步；否则抛弃. 对于下一步，所用步长或重复所抛弃的步所用的步长估计为 αh, 其中

$$c_{q+1}q!\alpha^{q+1}\left\|\frac{\nabla a_q}{\omega}\right\|_2 = \varepsilon.$$

如果采用这个步长，这个误差又正好与 h^{q+1} 成比例(即如果 ∇a_q 步步不变)，则下次试验正好满足.但是，∇a_q 不总是不变的，所以，总是使用稍微小的步长，以便在一定程度上可以期望试验 (9.13) 得到满足. 在后面所给的程序中，α 用

$$\alpha = \frac{1}{1.2}\left[\frac{\varepsilon}{c_{q+1}q!}\frac{1}{\left\|\frac{\nabla a_q}{\omega}\right\|_2}\right]^{1/q+1}$$

来估计. 还必须检验在其他诸阶的方法中能用的步长. 由于

$$\nabla^2 a_q \cong \frac{h^{q+2}}{q!}y^{(q+2)},$$

$$a_q \cong \frac{h^q}{q!}y^{(q)},$$

阶为 $q+1$ 和 $q-1$ 的方法能用的步长可估计为 αh, 其中

$$\alpha = \frac{1}{1.4}\left[\frac{\varepsilon}{c_{q+2}q!}\frac{1}{\left\|\frac{\nabla^2 a_q}{\omega}\right\|_2}\right]^{1/(q+2)} \qquad \text{对阶为 } q+1,$$

$$\alpha = \frac{1}{1.3}\left[\frac{\varepsilon}{c_q q!}\frac{1}{\left\|\frac{a_q}{\omega}\right\|_2}\right]^{1/q} \qquad \text{对阶为 } q-1.$$

因子 1.3 和 1.4 提供一个试验 (9.13) 能成功的大致范围. 它们的选择,强调要根据便于使用不变阶的方法,因变阶需要额外的机器时间,然后便于减阶,因为减阶每步只要耗费一点点工作量.

在下面情况下估计 α:

1. 如果一步失败,欲增加阶不可能,则估计 α.

2. 在上次改变阶或步长以后的 $q+1$ 步, 估计 α. [由 Nordsieck (1962) 所引用的试验以及由这位作者的证明表明:如果放大步长比这还频繁,就会积累大的误差,这样导致后面的步长必须减小. 还可参看本章末了的问题 2 和 3.]

3. 在上次估计 α 以后 10 步,如果步长不放大,则估计 α (可以减少过于频繁的试验).

估计 α 时,阶为对应于所选择的最大阶且适当改变步长. 若当前的阶为 q, 且 $\alpha \leqslant 1.1$, 下面给定的程序不放大步长,因为步长放大与试验所耗费的机器时间相比是不值得的.

下面的程序处理 N 个方程组且使用 Adams 或 Stiff 方法. Stiff 方法在第 11 章讨论.

开始值计算几乎是自动的. 由初值和微分方程能够计算 hy_0' 的值,这对于使用一阶方法是足够的. 然后阶控制程序能把阶增大到符合要求的程度.

$YMAX$ 参数是包含 (9.13) 所用的权的分量的数组. 在完成包含分量 y^i 的绝对值的一步以后修正 $YMAX$ 的元素,如果后者比 $YMAX(I)$ 的当前值大, 将 $YMAX$ 的元素进行修正. 这提供了在相对于 y^i 的过程中与最大值有关的误差试验,除非用户在每次调用之前改变 $YMAX$ 而不考虑这点. 所

估计的单步误差控制在参数 *EPS* 的范围内．因此，如果 f 与 y 无关，总的误差与步数成比例．如果方程是稳定的(即 $\partial f/\partial y$ 是负的)，则前面的误差减小，这样误差就比较小；如果方程是不稳定的，误差就比较大．这是因为这个误差控制仅仅根据局部截断误差而来．

程序注释．

应当注意，在 Stiff 方法的情形（Stiff 方法在第 11 章描述），调用程序 *MATINV* 来求矩阵的逆，实际上是用这个矩阵的逆左乘一向量．每次矩阵求逆的工作量近似为 $5N^3/6$ 个附加乘法，其中 N 是求解方程组中方程的个数．调用一级 Gauss 消去法程序能够代替单个调用子程序 *MATINV*，例如 Forsythe 和 Moler (1967) 第 68 页上的 *DECOMP*，而用二级(即回代过程) Gaussian 消去法程序能够代替用矩阵逆左乘向量的环节，例如 Forsythe 和 Morler (1967) 的第 69 页上的 *SOLVE*．这个环节出现在以语句标号 400 结束的五个语句中．这种改变引起的附加工作量为调用一次 *SOLVE* 的总工作量，调用次数大约为调用 *DECOMP* 或 *MATINV* 的 10 倍．对于任意大小的 N（大约 5 以上，这取决于所用的计算机），这种改变过的程序应当执行得更快．

表 9.2． 关于 Adams 方法误差对函数求值的关系

EPS	在 $t=20$ 与 e^{-20} 相比较的实际误差	函数求值次数	步数
0.10000D-01	0.10950D-01	201	73
0.10000D-02	0.20696D-02	262	88
0.10000D-03	−0.79186D-03	386	131
0.10000D-04	−0.35877D-04	391	132
0.10000D-05	−0.61224D-06	529	179
0.10000D-06	0.16620D-05	688	230
0.10000D-07	−0.45687D-08	773	259
0.10000D-08	0.14422D-07	734	247
0.10000D-09	0.94759D-08	767	261
0.10000D-10	0.20069D-08	946	323

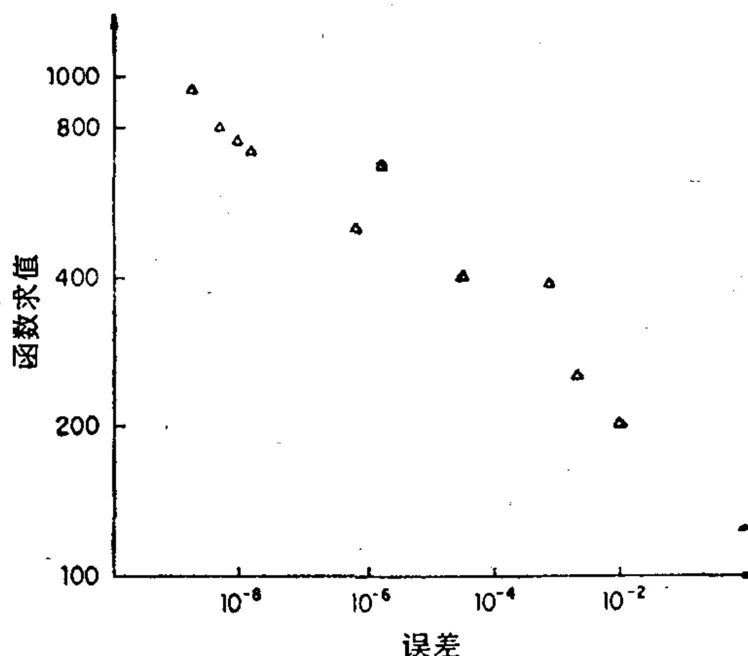


图 9.2　用自动的 Adams 方法对指数函数的误差

FORTRAN 程序

```
      SUBROUTINE DIFSUB(N,T,Y,SAVE,H,HMIN,HMAX,EPS,MF,YMAX,ERROR,KFLAG,
     1                  JSTART,MAXDER,PW)
      IMPLICIT REAL*8 (A-H,O-Z)
C***********************************************************************
C*                                                                     *
C* THIS SUBROUTINE INTEGRATES A SET OF N ORDINARY DIFFERENTIAL FIRST   *
C* ORDER EQUATIONS OVER ONE STEP OF LENGTH H AT EACH CALL.  H CAN BE   *
C* SPECIFIED BY THE USER FOR EACH STEP, BUT IT MAY BE INCREASED OR     *
C* DECREASED BY DIFSUB WITHIN THE RANGE HMIN TO HMAX IN ORDER TO       *
C* ACHIEVE AS LARGE A STEP AS POSSIBLE WHILE NOT COMMITTING A SINGLE   *
C* STEP ERROR WHICH IS LARGER THAN EPS IN THE L-2 NORM, WHERE EACH     *
C* COMPONENT OF THE ERROR IS DIVIDED BY THE COMPONENTS OF YMAX.        *
C*                                                                     *
C* THE PROGRAM REQUIRES THREE SUBROUTINES NAMED                        *
C*    DIFFUN(T,Y,DY)                                                   *
C*    MATINV(PW,N,M,J)                                                 *
C*    PEDERV(T,Y,PW,M)                                                 *
C* THE FIRST, DIFFUN, EVALUATES THE DERIVATIVES OF THE DEPENDENT       *
C* VARIABLES STORED IN Y(1,I) FOR I = 1 TO N, AND STORES THE           *
C* DERIVATIVES IN THE ARRAY DY.  THE SECOND IS CALLED ONLY IF THE      *
C* METHOD FLAG MF IS SET TO 1 OR 2 FOR STIFF METHODS.  IT MUST INVERT  *
C* THE N BY N MATRIX STORED IN THE ARRAY PW(M,M).  IF THE INVERSION IS *
C* SUCCESFUL, J SHOULD BE SET TO 1, OTHERWISE IT SHOULD BE SET TO -1.  *
C* PEDERV IS USED ONLY IF MF IS 1, AND COMPUTES THE PARTIAL            *
C* DERIVATIVES OF THE DIFFERENTIAL EQUATIONS AS DESCRIBED UNDER THE    *
C* MF PARAMETER.                                                       *
C*                                                                     *
C* THE PROGRAM USES DOUBLE PRECISION ARITHMETIC FOR ALL FLOATING       *
C* POINT VARIABLES EXCEPT THOSE STARTING WITH P.  THE FORMER ARE       *
C* SINGLE PRECISION TO SAVE TIME AND SPACE.                            *
C*                                                                     *
C* THE TEMPORARY STORAGE SPACE IS PROVIDED BY THE CALLER IN THE        *
C* SINGLE PRECISION ARRAY PW AND THE DOUBLE PRECISION ARRAY SAVE.      *
C* THE ARRAY PW IS USED ONLY TO HOLD THE MATRIX OF THE SAME NAME, BUT  *
C* SAVE IS USED TO HOLD SEVERAL ARRAYS.  THE REGIONS USED ARE          *
C*    SAVE(J,I)    1.LE.J.LE.8  AND  1.LE.I.LE.N  IS USED TO SAVE THE  *
C*                 VALUES OF Y IN CASE A STEP HAS TO BE REPEATED.      *
C*    SAVE(9,I)    IS USED MAINLY TO HOLD THE CORRECTION TERMS IN THE  *
C*                 CORRECTOR LOOP.                                     *
```

```
C*    SAVE(10,I)  IS USED TO SAVE THE VALUES OF THE SUMS OF ALL OF THE *
C*                CORRECTION TERMS IN THE PREVIOUS STEP AFTER THEY     *
C*                HAVE BEEN ACCUMULATED IN THE ARRAY ERROR IN THE      *
C*                CURRENT STEP.  THIS ENABLES THE BACKWARDS DIFFERENCE *
C*                OF ERROR TO BE FORMED.  IT IS USED TO ESTIMATE THE   *
C*                STEP SIZE FOR ONE ORDER HIGHER THAN CURRENT.         *
C*    SAVE(N1+I,1)  IS USED TO STORE THE DERIVATIVES WHEN THEY ARE     *
C*                COMPUTED BY DIFFUN.  IT IS ALSO ACCESSED AS          *
C*                SAVE(N2,1) AS A COMPLETE ARRAY.                      *
C*    SAVE(N5+I,1)  HOLDS THE DERIVATIVES DURING JACOBIAN EVALUATIONS. *
C*                IT IS REFERENCED AS SAVE(N6,1) AS A COMPLETE ARRAY.  *
C*                                                                     *
C* THE PARAMETERS TO THE SUBROUTINE DIFSUB HAVE                        *
C* THE FOLLOWING MEANINGS..                                            *
C*                                                                     *
C*   N        THE NUMBER OF FIRST ORDER DIFFERENTIAL EQUATIONS.  N     *
C*              MAY BE DECREASED ON LATER CALLS IF THE NUMBER OF       *
C*              ACTIVE EQUATIONS REDUCES, BUT IT MUST NOT BE           *
C*              INCREASED WITHOUT CALLING WITH JSTART = 0.             *
C*            THE INDEPENDENT VARIABLE.                                *
C*            AN 8 BY N ARRAY CONTAINING THE DEPENDENT VARIABLES AND   *
C*              THEIR SCALED DERIVATIVES.  Y(J+1,I) CONTAINS           *
C*              THE J-TH DERIVATIVE OF Y(I) SCALED BY                  *
C*              H**J/FACTORIAL(J) WHERE H IS THE CURRENT               *
C*              STEP SIZE. ONLY Y(1,I) NEED BE PROVIDED BY             *
C*              THE CALLING PROGRAM ON THE FIRST ENTRY.                *
C*                IF IT IS DESIRED TO INTERPOLATE TO NON MESH POINTS   *
C*              THESE VALUES CAN BE USED.  IF THE CURRENT STEP SIZE    *
C*              IS H AND THE VALUE AT T + E IS NEEDED, FORM            *
C*              S = E/H, AND THEN COMPUTE                              *
C*                              NQ                                     *
C*                Y(I)(T+E) =  SUM   Y(J+1,I)*S**J                     *
C*                             J=0                                     *
C*   SAVE     A BLOCK OF AT LEAST 12*N FLOATING POINT LOCATIONS        *
C*              USED BY THE SUBROUTINES.                               *
C*   H        THE STEP SIZE TO BE ATTEMPTED ON THE NEXT STEP.          *
C*              H MAY BE ADJUSTED UP OR DOWN BY THE PROGRAM            *
C*              IN ORDER TO ACHEIVE AN ECONOMICAL INTEGRATION.         *
C*              HOWEVER, IF THE H PROVIDED BY THE USER DOES            *
C*              NOT CAUSE A LARGER ERROR THAN REQUESTED, IT            *
C*              WILL BE USED.  TO SAVE COMPUTER TIME, THE USER IS      *
C*              ADVISED TO USE A FAIRLY SMALL STEP FOR THE FIRST       *
C*              CALL.  IT WILL BE AUTOMATICALLY INCREASED LATER.       *
C*   HMIN     THE MINIMUM STEP SIZE THAT WILL BE USED FOR THE          *
C*              INTEGRATION.  NOTE THAT ON STARTING THIS MUST          *
C*              MUCH SMALLER THAN THE AVERAGE H EXPECTED SINCE         *
C*              A FIRST ORDER METHOD IS USED INITIALLY.                *
C*   HMAX     THE MAXIMUM SIZE TO WHICH THE STEP WILL BE INCREASED     *
C*   EPS      THE ERROR TEST CONSTANT.  SINGLE STEP ERROR ESTIMATES    *
C*              DIVIDED BY  YMAX(I)  MUST BE LESS THAN THIS            *
C*              IN THE EUCLIDEAN NORM. THE STEP AND/OR ORDER IS        *
C*              ADJUSTED TO ACHEIVE THIS.                              *
C*   MF       THE METHOD INDICATOR. THE FOLLOWING ARE ALLOWED..        *
C*                     0   AN ADAMS PREDICTOR CORRECTOR IS USED.       *
C*                     1   A MULTI-STEP METHOD SUITABLE FOR STIFF      *
C*                             SYSTEMS IS USED. IT WILL ALSO WORK FOR  *
C*                             NON STIFF SYSTEMS.  HOWEVER THE USER    *
C*                             MUST PROVIDE A SUBROUTINE PEDERV WHICH  *
C*                             EVALUATES THE PARTIAL DERIVATIVES OF    *
C*                             THE DIFFERENTIAL EQUATIONS WITH RESPECT *
C*                             TO THE Y'S.   THIS IS DONE BY CALL      *
C*                             PEDERV(T,Y,PW,M). PW IS AN N BY N ARRAY *
C*                             WHICH MUST BE SET TO THE PARTIAL OF     *
C*                             THE I-TH EQUATION WITH RESPECT          *
C*                             TO THE J DEPENDENT VARIABLE IN PW(I,J). *
C*                             PW IS ACTUALLY STORED IN AN M BY M      *
C*                             ARRAY WHERE M IS THE VALUE OF N USED ON *
C*                             THE FIRST CALL TO THIS PROGRAM.         *
```

```
C*                    2   THE SAME AS CASE 1, EXCEPT THAT THIS          *
C*                            SUBROUTINE COMPUTES THE PARTIAL            *
C*                            DERIVATIVES BY NUMERICAL DIFFERENCING      *
C*                            OF THE DERIVATIVES. HENCE PEDERV IS        *
C*                            NOT CALLED.                                *
C*   YMAX    AN ARRAY OF  N LOCATIONS WHICH CONTAINS THE MAXIMUM         *
C*              OF EACH Y SEEN SO FAR.  IT SHOULD NORMALLY BE SET TO     *
C*              1 IN EACH COMPONENT BEFORE THE FIRST ENTRY. (SEE THE     *
C*              DESCRIPTION OF EPS.)                                     *
C*   ERROR   AN ARRAY OF  N ELEMENTS WHICH CONTAINS THE ESTIMATED        *
C*              ONE STEP ERROR IN EACH COMPONENT.                        *
C*   KFLAG   A COMPLETION CODE WITH THE FOLLOWING MEANINGS..             *
C*                   +1  THE STEP WAS SUCCESFUL.                         *
C*                   -1  THE STEP WAS TAKEN WITH H = HMIN, BUT THE       *
C*                          REQUESTED ERROR WAS NOT ACHIEVED.            *
C*                   -2  THE MAXIMUM ORDER SPECIFIED WAS FOUND TO        *
C*                          BE TOO LARGE.                                *
C*                   -3  CORRECTOR CONVERGENCE COULD NOT BE              *
C*                          ACHIEVED FOR H .GT. HMIN.                    *
C*                   -4  THE REQUESTED ERROR IS SMALLER THAN CAN         *
C*                          BE HANDLED FOR THIS PROBLEM.                 *
C*   JSTART  AN INPUT INDICATOR WITH THE FOLLOWING MEANINGS..            *
C*                   -1  REPEAT THE LAST STEP WITH A NEW H               *
C*                    0  PERFORM THE FIRST STEP.  THE FIRST STEP         *
C*                          MUST BE DONE WITH THIS VALUE OF JSTART       *
C*                          SO THAT THE SUBROUTINE CAN INITIALIZE        *
C*                          ITSELF.                                      *
C*                   +1  TAKE A NEW STEP CONTINUING FROM THE LAST.       *
C*              JSTART IS SET TO NQ, THE CURRENT ORDER OF THE METHOD     *
C*              AT EXIT.  NQ IS ALSO THE ORDER OF THE MAXIMUM            *
C*              DERIVATIVE AVAILABLE.                                    *
C*   MAXDER  THE MAXIMUM DERIVATIVE THAT SHOULD BE USED IN THE           *
C*              METHOD.  SINCE THE ORDER IS EQUAL TO THE HIGHEST         *
C*              DERIVATIVE USED, THIS RESTRICTS THE ORDER. IT MUST       *
C*              BE LESS THAN 8 FOR ADAMS AND 7 FOR STIFF METHODS.        *
C*   PW      A BLOCK OF AT LEAST N**2 FLOATING POINT LOCATIONS.          *
C**********************************************************************
      DIMENSION Y(8,1),YMAX(1),SAVE(10,1),ERROR(1),PW(1),
     1           A(8),PERTST(7,2,3)
C**********************************************************************
C* THE COEFFICIENTS IN PERTST ARE USED IN SELECTING THE STEP AND       *
C* ORDER, THEREFORE ONLY ABOUT ONE PERCENT ACCURACY IS NEEDED.         *
C**********************************************************************
      DATA PERTST /2.0,4.5,7.333,10.42,13.7,17.15,1.0,
     1             2.0,12.0,24.0,37.89,53.33,70.08,87.97,
     2             3.0,6.0,9.167,12.5,15.98,1.0,1.0,
     3             12.0,24.0,37.89,53.33,70.08,87.97,1.0,
     4             1.,1.,0.5,0.1667,0.04133,0.008267,1.0,
     5             1.0,1.0,2.0,1.0,.3157,.07407,.01397/
      DATA A(2) / -1.0 /
      IRET = 1
      KFLAG = 1
      IF (JSTART.LE.0) GO TO 140
C**********************************************************************
C* BEGIN BY SAVING INFORMATION FOR POSSIBLE RESTARTS AND CHANGING      *
C* H BY THE FACTOR R IF THE CALLER HAS CHANGED H.  ALL VARIABLES       *
C* DEPENDENT ON H MUST ALSO BE CHANGED.                                *
C* E IS A COMPARISON FOR ERRORS OF THE CURRENT ORDER NQ. EUP IS        *
C* TO TEST FOR INCREASING THE ORDER, EDWN FOR DECREASING THE ORDER.    *
C* HNEW IS THE STEP SIZE THAT WAS USED ON THE LAST CALL.               *
C**********************************************************************
 100  DO 110 I = 1,N
         DO 110 J = 1,K
 110        SAVE(J,I) = Y(J,I)
      HOLD = HNEW
      IF (H.EQ.HOLD) GO TO 130
 120  RACUM = H/HOLD
```

```
      IRET1 = 1
      GO TO 750
  130 NQOLD = NQ
      TOLD = T
      RACUM = 1.0
      IF (JSTART.GT.0) GO TO 250
      GO TO 170
  140 IF (JSTART.EQ.-1) GO TO 160
C***********************************************************************
C* ON THE FIRST CALL, THE ORDER IS SET TO 1 AND THE INITIAL            *
C* DERIVATIVES ARE CALCULATED.                                         *
C***********************************************************************
      NQ = 1
      N3 = N
      N1 = N*10
      N2 = N1 + 1
      N4 = N**2
      N5 = N1 + N
      N6 = N5 + 1
      CALL DIFFUN(T,Y,SAVE(N2,1))
      DO 150 I = 1,N
  150   Y(2,I) = SAVE(N1+I,1)*H
      HNEW = H
      K = 2
      GO TO 100
C***********************************************************************
C* REPEAT LAST STEP BY RESTORING SAVED INFORMATION.                    *
C***********************************************************************
  160 IF (NQ.EQ.NQOLD) JSTART = 1
      T = TOLD
      NQ = NQOLD
      K = NQ + 1
      GO TO 120
C***********************************************************************
C* SET THE COEFFICIENTS THAT DETERMINE THE ORDER AND THE METHOD        *
C* TYPE.  CHECK FOR EXCESSIVE ORDER.  THE LAST TWO STATEMENTS OF       *
C* THIS SECTION  SET IWEVAL .GT.0 IF PW IS TO BE RE-EVALUATED          *
C* BECAUSE OF THE ORDER CHANGE, AND THEN REPEAT THE INTEGRATION        *
C* STEP IF IT HAS NOT YET BEEN DONE (IRET = 1) OR SKIP TO A FINAL      *
C* SCALING BEFORE EXIT IF IT HAS BEEN COMPLETED (IRET = 2).            *
C***********************************************************************
  170 IF (MF.EQ.0) GO TO 180
      IF (NQ.GT.6) GO TO 190
      GO TO (221,222,223,224,225,226),NQ
  180 IF (NQ.GT.7) GO TO 190
      GO TO (211,212,213,214,215,216,217),NQ
  190 KFLAG = -2
      RETURN
C***********************************************************************
C* THE FOLLOWING COEFFICIENTS SHOULD BE DEFINED TO THE MAXIMUM         *
C* ACCURACY PERMITTED BY THE MACHINE.  THEY ARE, IN THE ORDER USED..   *
C*                                                                     *
C* -1                                                                  *
C* -1/2,-1/2                                                           *
C* -5/12,-3/4,-1/6                                                     *
C* -3/8,-11/12,-1/3,-1/24                                              *
C* -251/720,-25/24,-35/72,-5/48,-1/120                                 *
C* -95/288,-137/120,-5/8,-17/96,-1/40,-1/720                           *
C* -19087/60480,-49/40,-203/270,-49/192,-7/144,-7/1440,-1/5040         *
C*                                                                     *
C* -1                                                                  *
C* -2/3,-1/3                                                           *
C* -6/11,-6/11,-1/11                                                   *
C* -12/25,-7/10,-1/5,-1/50                                             *
C* -120/274,-225/274,-85/274,-15/274,-1/274                            *
C* -180/441,-58/63,-15/36,-25/252,-3/252,-1/1764                       *
C***********************************************************************
  211 A(1) = -1.0
```

```
      GO TO 230
212   A(1) = -0.500000000
      A(3) = -0.500000000
      GO TO 230
213   A(1) = -0.4166666666666667
      A(3) = -0.750000000
      A(4) = -0.1666666666666667
      GO TO 230
214   A(1) = -0.375000000
      A(3) = -0.9166666666666667
      A(4) = -0.3333333333333333
      A(5) = -0.0416666666666667
      GO TO 230
215   A(1) = -0.3486111111111111
      A(3) = -1.0416666666666667
      A(4) = -0.4861111111111111
      A(5) = -0.1041666666666667
      A(6) = -0.0083333333333333
      GO TO 230
216   A(1) = -0.3298611111111111
      A(3) = -1.1416666666666667
      A(4) = -0.625000000
      A(5) = -0.1770833333333333
      A(6) = -0.0250000000
      A(7) = -0.0013888888888889
      GO TO 230
217   A(1) = -0.3155919312169312
      A(3) = -1.235000000
      A(4) = -0.7518518518518519
      A(5) = -0.2552083333333333
      A(6) = -0.0486111111111111
      A(7) = -0.0048611111111111
      A(8) = -0.0001984126984126984
      GO TO 230
221   A(1) = -1.000000000
      GO TO 230
222   A(1) = -0.6666666666666667
      A(3) = -0.3333333333333333
      GO TO 230
223   A(1) = - 0.5454545454545455
      A(3) = A(1)
      A(4) = -0.0909090909090909091
      GO TO 230
224   A(1) = -0.480000000
      A(3) = -0.700000000
      A(4) = -0.200000000
      A(5) = -0.020000000
      GO TO 230
225   A(1) = -0.437956204379562
      A(3) = -0.8211678832116798
      A(4) = -0.3102189781021898
      A(5) = -0.0547445255474452б
      A(6) = -0.00364963503649635О4
      GO TO 230
226   A(1) = -0.4081632653061225
      A(3) = -0.9206349206349206
      A(4) = -0.4166666666666667
      A(5) = -0.0992063492063492
      A(6) = -0.0119047619047619
      A(7) = -0.0005668934240362В2
230   K = NQ+1
      IDOUB = K
      MTYP = (4 - MF)/2
      ENQ2 = .5/FLOAT(NQ + 1)
      ENQ3 = .5/FLOAT(NQ + 2)
      ENQ1 = 0.5/FLOAT(NQ)
      PEPSH = EPS
      EUP = (PERTST(NQ,MTYP,2)*PEPSH)**2
```

```
      E = (PERTST(NQ,MTYP,1)*PEPSH)**2
      EDWN =(PERTST(NQ,MTYP,3)*PEPSH)**2
      IF (EDWN.EQ.0) GO TO 780
      BND = EPS*ENQ3/DFLOAT(N)
 240  IWEVAL = MF
      GO TO ( 250 , 680 ),IRET
C**********************************************************************
C* THIS SECTION COMPUTES THE PREDICTED VALUES BY EFFECTIVELY          *
C* MULTIPLYING THE SAVED INFORMATION BY THE PASCAL TRIANGLE           *
C* MATRIX.                                                            *
C**********************************************************************
 250  T = T + H
      DO 260 J = 2,K
        DO 260 J1 = J,K
          J2 = K - J1 + J - 1
          DO 260 I = 1,N
 260        Y(J2,I) = Y(J2,I) + Y(J2+1,I)
C**********************************************************************
C*   UP TO 3 CORRECTOR ITERATIONS ARE TAKEN.  CONVERGENCE IS TESTED   *
C*   BY REQUIRING CHANGES TO BE LESS THAN BND WHICH IS DEPENDENT ON   *
C*   THE ERROR TEST CONSTANT.                                         *
C*      THE SUM OF THE CORRECTIONS IS ACCUMULATED IN THE ARRAY        *
C*   ERROR(I).  IT IS EQUAL TO THE K-TH DERIVATIVE OF Y MULTIPLIED    *
C*   BY  H**K/(FACTORIAL(K-1)*A(K)), AND IS THEREFORE PROPORTIONAL    *
C*   TO THE ACTUAL ERRORS TO THE LOWEST POWER OF H PRESENT. (H**K)    *
C**********************************************************************
      DO 270 I = 1,N
 270    ERROR(I) = 0.0
      DO 430 L = 1,3
        CALL DIFFUN (T,Y,SAVE(N2,1))
C**********************************************************************
C*   IF THERE HAS BEEN A CHANGE OF ORDER OR THERE HAS BEEN TROUBLE    *
C*   WITH CONVERGENCE, PW IS RE-EVALUATED PRIOR TO STARTING THE       *
C*   CORRECTOR ITERATION IN THE CASE OF STIFF METHODS.  IWEVAL IS     *
C*   THEN SET TO -1 AS AN INDICATOR THAT IT HAS BEEN DONE.            *
C**********************************************************************
      IF (IWEVAL.LT.1) GO TO 350
        IF (MF.EQ.2) GO TO 310
        CALL PEDERV(T,Y,PW,N3)
        R = A(L)*H
        DO 280 I = 1,N4
 280      PW(I) = PW(I)*R-
C**********************************************************************
C* ADD THE IDENTITY MATRIX TO THE JACOBIAN AND INVERT TO GET PW.      *
C**********************************************************************
 290    DO 300 I = 1,N
 300      PW(I*(N3+1)-N3) = 1.0 + PW(I*(N3+1)-N3)
        IWEVAL = -1
        CALL MATINV(PW,N,N3,J1)
        IF (J1.GT.0) GO TO 350
        GO TO 440
C**********************************************************************
C* EVALUATE THE JACOBIAN INTO PW BY NUMERICAL DIFFERENCING.  R IS THE *
C* CHANGE MADE TO THE ELEMENT OF Y.  IT IS EPS RELATIVE TO Y WITH     *
C* A MINIMUM OF EPS**2.                                               *
C**********************************************************************
 310    DO 320 I = 1,N
 320      SAVE(9,I) = Y(1,I)
        DO 340 J = 1,N
          R = EPS*DMAX1(EPS,DABS(SAVE(9,J)))
          Y(1,J) = Y(1,J) + R
          D = A(1)*H/R
          CALL DIFFUN(T,Y,SAVE(N6,1))
          DO 330 I = 1,N
 330        PW(I+(J-1)*N3) = (SAVE(N5+I,1) - SAVE(N1+I,1))*D
 340      Y(1,J) = SAVE(9,J)
        GO TO 290
```

```
 350      IF (MF.NE.0) GO TO 370
          DO 360 I = 1,N
 360        SAVE(9,I) = Y(2,I) - SAVE(N1+I,1)*H
          GO TO 410
 370      DO 380 I = 1,N
 380        SAVE(N5+I,1) = Y(2,I) - SAVE(N1+I,1)*H
          DO 400 I = 1,N
            D = 0.0
            DO 390 J = 1,N
 390          D = D + PW(I+(J-1)*N3)*SAVE(N5+J,1)
 400        SAVE(9,I) = D
 410      NT = N
C***********************************************************************
C* CORRECT AND SEE IF ALL CHANGES ARE LESS THAN BND RELATIVE TO YMAX.  *
C* IF SO, THE CORRECTOR IS SAID TO HAVE CONVERGED.                     *
C***********************************************************************
        DO 420 I = 1,N
          Y(1,I) = Y(1,I) + A(1)*SAVE(9,I)
          Y(2,I) = Y(2,I) - SAVE(9,I)
          ERROR(I) = ERROR(I) + SAVE(9,I)
          IF  (DABS(SAVE(9,I)).LE.(BND*YMAX(I))) NT = NT - 1
 420      CONTINUE
        IF (NT.LE.0) GO TO 490
 430    CONTINUE
C***********************************************************************
C* THE CORRECTOR ITERATION FAILED TO CONVERGE IN 3 TRIES.  VARIOUS     *
C* POSSIBILITIES ARE CHECKED FOR.  IF H IS ALREADY HMIN AND            *
C* THIS IS EITHER ADAMS METHOD OR THE STIFF METHOD IN WHICH THE        *
C* MATRIX PW HAS ALREADY BEEN RE-EVALUATED, A NO CONVERGENCE EXIT      *
C* IS TAKEN. OTHERWISE THE MATRIX PW IS RE-EVALUATED AND/OR THE        *
C* STEP IS REDUCED TO TRY AND GET CONVERGENCE.                         *
C***********************************************************************
 440  T = TOLD
      IF ((H.LE.(HMIN*1.00001)).AND.((IWEVAL - MTYP).LT.-1)) GO TO 460
      IF ((MF.EQ.0).OR.(IWEVAL.NE.0)) RACUM = RACUM*0.25D0
      IWEVAL = MF
      IRET1 = 2
      GO TO 750
 460  KFLAG = -3
 470  DO 480 I = 1,N
        DO 480 J = 1,K
 480      Y(J,I) = SAVE(J,I)
      H = HOLD
      NQ = NQOLD
      JSTART = NQ
      RETURN
C***********************************************************************
C* THE CORRECTOR CONVERGED AND CONTROL IS PASSED TO STATEMENT 520      *
C* IF THE ERROR TEST IS O.K.,  AND TO 540 OTHERWISE.                   *
C* IF THE STEP IS O.K. IT IS ACCEPTED. IF IDOUB HAS BEEN REDUCED       *
C* TO ONE,  A TEST IS MADE TO SEE IF THE STEP CAN BE INCREASED         *
C* AT THE CURRENT ORDER OR BY GOING TO ONE HIGHER OR ONE LOWER.        *
C* SUCH A CHANGE IS ONLY MADE IF THE STEP CAN BE INCREASED BY AT       *
C* LEAST 1.1.  IF NO CHANGE IS POSSIBLE IDOUB IS SET TO 10 TO          *
C* PREVENT FUTHER TESTING FOR 10 STEPS.                                *
C* IF A CHANGE IS POSSIBLE, IT IS MADE AND IDOUB IS SET TO             *
C* NQ + 1    TO PREVENT FURTHER TESING FOR THAT NUMBER OF STEPS.       *
C* IF THE ERROR WAS TOO LARGE, THE OPTIMUM STEP SIZE FOR THIS OR       *
C* LOWER ORDER IS COMPUTED, AND THE STEP RETRIED.  IF IT SHOULD        *
C* FAIL TWICE MORE IT IS AN INDICATION THAT THE DERIVATIVES THAT       *
C* HAVE ACCUMULATED IN THE Y ARRAY HAVE ERRORS OF THE WRONG ORDER      *
C* SO THE FIRST DERIVATIVES ARE RECOMPUTED AND THE ORDER IS SET        *
C* TO 1.                                                               *
C***********************************************************************
 490  D = 0.0
      DO 500 I = 1,N
 500    D = D + (ERROR(I)/YMAX(I))**2
      IWEVAL = 0
```

```
      IF (D.GT.E) GO TO 540
      IF (K.LT.3) GO TO 520
C*********************************************************************
C* COMPLETE THE CORRECTION OF THE HIGHER ORDER DERIVATIVES AFTER A   *
C* SUCCESFUL STEP.                                                   *
C*********************************************************************
      DO 510 J = 3,K
        DO 510 I = 1,N
 510      Y(J,I) = Y(J,I) + A(J)*ERROR(I)
 520  KFLAG = +1
      HNEW = H
      IF (IDOUB.LE.1) GO TO 550
      IDOUB = IDOUB - 1
      IF (IDOUB.GT.1) GO TO 700
      DO 530 I = 1,N
 530    SAVE(10,I) = ERROR(I)
      GO TO 700
C*********************************************************************
C* REDUCE THE FAILURE FLAG COUNT TO CHECK FOR MULTIPLE FAILURES.     *
C* RESTORE T TO ITS ORIGINAL VALUE AND TRY AGAIN UNLESS THERE HAVE   *
C* THREE FAILURES.  IN THAT CASE THE DERIVATIVES ARE ASSUMED TO HAVE *
C* ACCUMULATED ERRORS SO A RESTART FROM THE CURRENT VALUES OF Y IS   *
C* TRIED.                                                            *
C*********************************************************************
 540  KFLAG = KFLAG - 2
      IF (H.LE.(HMIN*1.00001)) GO TO 740
      T = TOLD
      IF (KFLAG.LE.-5) GO TO 720
C*********************************************************************
C* PR1, PR2, AND PR3 WILL CONTAIN THE AMOUNTS BY WHICH THE STEP SIZE *
C* CHOULD BE DIVIDED AT ORDER ONE LOWER, AT THIS ORDER, AND AT ORDER *
C* ONE HIGHER RESPECTIVELY.                                          *
C*********************************************************************
 550  PR2 = (D/E)**ENQ2*1.2
      PR3 = 1.E+20
      IF ((NQ.GE.MAXDER).OR.(KFLAG.LE.-1)) GO TO 570
      D = 0.0
      DO 560 I = 1,N
 560    D = D + ((ERROR(I) - SAVE(10,I))/YMAX(I))**2
      PR3 = (D/EUP)**ENQ3*1.4
 570  PR1 = 1.E+20
      IF (NQ.LE.1) GO TO 590
      D = 0.0
      DO 580 I = 1,N
 580    D = D + (Y(K,I)/YMAX(I))**2
      PR1 = (D/EDWN)**ENQ1*1.3
 590  CONTINUE
      IF (PR2.LE.PR3) GO TO 650
      IF (PR3.LT.PR1) GO TO 660
 600  R = 1.0/AMAX1(PR1,1.E-4)
      NEWQ = NQ - 1
 610  IDOUB = 10
      IF ((KFLAG.EQ.1).AND.(R.LT.(1.1))) GO TO 700
      IF (NEWQ.LE.NQ) GO TO 630
C*********************************************************************
C* COMPUTE ONE ADDITIONAL SCALED DERIVATIVE IF ORDER IS INCREASED.   *
C*********************************************************************
      DO 620 I = 1,N
 620    Y(NEWQ+1,I) = ERROR(I)*A(K)/DFLOAT(K)
 630  K = NEWQ + 1
      IF (KFLAG.EQ.1) GO TO 670
      RACUM = RACUM*R
      IRET1 = 3
      GO TO 750
 640  IF (NEWQ.EQ.NQ) GO TO 250
      NQ = NEWQ
      GO TO 170
 650  IF (PR2.GT.PR1) GO TO 600
```

```
      NEWQ = NQ
      R = 1.0/AMAX1(PR2,1.E-4)
      GO TO 610
660   R = 1.0/AMAX1(PR3,1.E-4)
      NEWQ = NQ + 1
      GO TO 610
670   IRET = 2
      R = DMIN1(R,HMAX/DABS(H))
      H = H*R
      HNEW = H
      IF (NQ.EQ.NEWQ) GO TO 680
      NQ = NEWQ
      GO TO 170
680   R1 = 1.0
      DO 690 J = 2,K
        R1 = R1*R
        DO 690 I = 1,N
690       Y(J,I) = Y(J,I)*R1
      IDOUB = K
700   DO 710 I = 1,N
710     YMAX(I) = DMAX1(YMAX(I),DABS(Y(1,I)))
      JSTART = NQ
      RETURN
720   IF (NQ.EQ.1) GO TO 780
      CALL DIFFUN (T,Y,SAVE(N2,1))
      R = H/HOLD
      DO 730 I = 1,N
        Y(1,I) = SAVE(1,I)
        SAVE(2,I) = HOLD*SAVE(N1+I,1)
730     Y(2,I) = SAVE(2,I)*R
      NQ = 1
      KFLAG = 1
      GO TO 170
740   KFLAG = -1
      HNEW = H
      JSTART = NQ
      RETURN
C*********************************************************************
C* THIS SECTION SCALES ALL VARIABLES CONNECTED WITH H AND RETURNS    *
C* TO THE ENTERING SECTION.                                          *
C*********************************************************************
750   RACUM = DMAX1(DABS(HMIN/HOLD),RACUM)
      RACUM = DMIN1(RACUM,DABS(HMAX/HOLD))
      R1 = 1.0
      DO 760 J = 2,K
        R1 = R1*RACUM
        DO 760 I = 1,N
760       Y(J,I) = SAVE(J,I)*R1
      H = HOLD*RACUM
      DO 770 I = 1,N
770     Y(1,I) = SAVE(1,I)
      IDOUB = K
      GO TO ( 130 , 250 , 640 ),IRET
780   KFLAG = -4
      GO TO 470
      END
```

用这个程序自 $y(0)=1$ 到 $t=20$，对于 $E=EPS=10^{-i}$ $(i=2, 3\cdots, 11)$ 积分 $y'=-y$，结果表示在前面的表9.2和图9.2中. $YMAX(1)$ 在每步之前取 y 的当前值.

问　　题

1. 如果由方程(9.13)定义的截断误差是 $\mathbf{d}_n$，证明经 T 变换后，方程的截断误差是 $T\mathbf{d}_n$.
2. 利用 $\mathbf{a}=[y, hy', h^2y''/2]^T$ 的表示形式，用

$$A=\begin{bmatrix}1 & 1 & 1\\ 0 & 1 & 2\\ 0 & 0 & 1\end{bmatrix},\quad l=\begin{bmatrix}\frac{5}{12}\\ 1\\ \frac{1}{2}\end{bmatrix}$$

   给出三阶二步 Adams-Bashforth-Moulton 方法. 在这种表示形式中，方程 $y'=0$ 的稳定性矩阵如何?

   如果在第 $2i(i=1, 2, \cdots)$ 步之后用量 α_i 改变步长，那么用初值表示的解如何? 证明: 结果与所用的步长无关. 如果每步后均改变步长，这是否还真确?
3. 对四阶的三步 Adams-Bashforth-Moulton 方法重复问题2. 必须积分多少步改变一次步长，才能使结果与所用的步长无关?
4. 对在步长 h 以后取步长为 αh 的情形，推导形式为

$$y(t+\alpha h)=y(t)+h\beta_1 y'(t)+h\beta_2 y'(t-h)+O(h^3)$$

$$y(t+\alpha h)=y(t)+h\beta_0^* y'(t+\alpha h)+h\beta_1^* y'(t)+h\beta_2^* y'(t-h)+O(h^4)$$

   的"预估"公式及"校正"公式. 把它与二步 Adams-Bashforth-Moulton 积分公式的内插步长变化方法进行比较. 证明: 用变步长执行一步的结果在两种方法中是一样的.
5. 证明:三步 Adams-Bashforth-Moulton 方法的矩阵 B 与校正公式的阶(3 或 4)无关. 如果使用二阶校正公式，B 一样吗?
6. 方程(9.13)给出所谓 $P3C4$ 方法的三阶预估与四阶校正公式. 用同样的表示形式，$P3C4$ Adams 方法的矩阵和向量如何?

10. 多值方法的存在性、收敛性和误差估计

在这一章中，我们研究多步方法和多值方法的理论基础. 一般只研究单个方程的情形，然而结果同样可应用于方程组，甚至方程组中各个方程的阶不同而且对每个方程使用适合于它的阶的方法进行积分的情形也一样.

定义了收敛性和稳定性之后，我们证明由稳定性推出根条件，由根条件与阶至少为 1（称作相容性）一起推出收敛性，由收敛性推出根条件，由根条件和相容性推出稳定性. 对多步法还要证明由收敛性推出相容性. 这样完成了如图 10.1 所示的对一阶方程的关系链.



图 10.1　基本定理

多值方法对高阶方程的推广在第 9 章中讨论了. 这一章考虑形如

$$y^{(p)}=f(y,y',\cdots,y^{(p-q)},t)\qquad 1\leqslant q\leqslant p \quad (10.1)$$

的 p 阶方程，即最高的 $q-1$ 个导数不出现（当然 p 阶导数除外）. 我们将称它为 q 微分的 p 阶方程. 若 $q=1$，我们称它为一般的 p 阶方程. 若 $q=p$，称它为特殊的 p 阶方程. 特殊的二阶方程 $y''=f(y,t)$ 经常出现在天体运动的保守力学系统中，所以值得专门研究.

类似于图 10.1 中的那些关系链对 p 阶方程也成立。方法的阶这样定义，使得为了相容性阶数至少必须为 p。[这不同于 Henrici (1962) 按量 $p-1$ 的定义。我们称具有局部误差 $O(h^{r+1})$ 的方法为 r 阶方法；但他称它为 $(r+1-p)$ 阶方法] 对 q 微分方程，我们定义 q 稳定性，q 收敛性和 q 根条件的概念，然后将看到，图 10.1 除了收敛性 $\Longrightarrow$ 相容性的结果还没有证明外，都是成立的。

我们将证明 Dahlquist 的重要定理，它限制了对于一阶方程多步方法的稳定阶，然后证明最大阶稳定的多值方法的存在性。为了得到稳定性，向量 $\mathbf{l}$ 的选取有一些自由度，它可以用来选取误差的系数。我们考察这个问题，目的是看看能得到什么样的误差界及误差估计。

将对固定阶和固定步长的方法得到这些结果。固定步长的结果推广到变步长(见 4.6 节)或者甚至推广到变阶方法是简单的。但对多值方法作同样的推广是困难的，并有很多限制，而且在实际中常用的大多数自动方法还缺乏适当的理论作为基础。然而有确实的理由希望这些结果会得到推广，而且如所想象的，自动方法与固定步长和固定阶方法有类似的结果。

有些结果容易用通常的多步公式来表示和证明，而另一些结果用对多值方法所提出的矩阵符号更容易处理。p 阶方程 (10.1) 可以用形如

$$\sum_{i=0}^{k}\left(\alpha_i y_{n-i}+\beta_i \frac{h^p}{p!} f_{n-i}\right)=0 \tag{10.2}$$

的多步方法来积分，其中 $f_m=f(y_m, y'_m, \cdots, y_m^{(p-1)}, t_m)$，并且 $y'_m, \cdots, y_m^{(p-1)}$ 的值可由对 $0<q<p$ 的公式

$$\frac{h^q y_m^{(q)}}{q!}=\sum_{i=0}^{k}\left(\alpha_{qi} y_{m-i}+\beta_{qi} \frac{h^p}{p!} f_{m-i}\right) \tag{10.3}$$

得到。如果 β_0 或 β_{q0} 不为零，这些公式是隐式的。

由于计算 (10.3) 的值的附加工作量，这个方法除了对于特殊的 p 阶方程外，通常是不用的．对特殊的 p 阶方程，只要用 (10.2)．

在不需要 (10.3) 的情形，可以用矩阵符号来表示．如果先应用 (10.2) 作为预估公式 ($\beta_0 = 0$ 和 $\alpha_0 = -1$)，然后用来作为校正公式，按形式

$$y_{n,(m+1)} = \sum_{i=1}^{k}\left(\alpha_i^* y_{n-i} + \beta_i^* \frac{h^p}{p!} f_{n-i}\right) + \beta_0^* \frac{h^p}{p!} f(y_{n,(m)}, t_n) \tag{10.4}$$

进行迭代，我们可以定义向量 $\mathbf{y}_n = [y_n, y_{n-1}, \cdots, y_{n-k+1}, (h^p/p!)y_n^{(p)}, \cdots, (h^p/p!)y_{n-k+1}^{(p)}]^T$，并将方法表示成

$$\begin{aligned} \mathbf{y}_{n,(0)} &= B\mathbf{y}_{n-1}, \\ \mathbf{y}_{n,(m+1)} &= \mathbf{y}_{n,(m)} + \mathbf{c}G(\mathbf{y}_{n,(m)}), \end{aligned} \tag{10.5}$$

其中

$$B = \left[\begin{array}{ccc|ccc} \alpha_1 & \alpha_2 \ \cdots & \alpha_k & \beta_1 & \cdots & \beta_k \\ 1 & & & & & \\ & \ddots & & & 0 & \\ & 1 & 0 & & & \\ \hline \nu_1 & \cdots & \nu_k & \delta_1 & \cdots & \delta_k \\ & & & 1 & & \\ & 0 & & & \ddots & \\ & & & & 1 & 0 \end{array}\right], \qquad \mathbf{c} = \left[\begin{array}{c} \beta_0^* \\ 0 \\ \vdots \\ 0 \\ \hline 0 \\ \vdots \\ 0 \end{array}\right],$$

$$\nu_i = \frac{\alpha_i - \alpha_i^*}{\beta_0^*},$$

$$\delta_i = \frac{\beta_i - \beta_i^*}{\beta_0^*},$$

$$G(\mathbf{y}_{n,(m)}) = \frac{h^p}{p!} f(y_{n,(m)}, t_n) - \frac{h^p}{p!} y_{n,(m)}^{(p)}.$$

稳定性分析与一阶方法等同．稳定性由矩阵

$$S_n = \prod_{i=0}^{M-1}\left(I + \mathbf{c}\frac{\partial G}{\partial \mathbf{y}}(\boldsymbol{\xi}_i)\right) B \tag{10.6}$$

的性质来确定．在 $h \to 0$ 时，

$$S = (I - \mathbf{c}\boldsymbol{\delta}_K^T)^{MB} \tag{10.7}$$

只要涉及到向量 $\mathbf{y}$，我们将讨论多步方法的这种表示形式．

应用 § 9.2 描述的变换，可以导出等价的方法．特别将用其中 $\mathbf{a} = [y, hy', \cdots, h^{k-1}y^{(k-1)}/(k-1)!]^T$ 的多值公式．在这一章中，除非另加说明，$\mathbf{a}$ 总是指这种特殊形式，并且称它为多值方法的标准形式．不另外指出时，我们假定在 $\mathbf{a}$ 中有 k 个分量．

更一般的 p 阶方程 (10.1)，可以直接按这种公式处理，因为导数均可取用，所以没有必要给出象 (10.3) 的另外的公式计算它们．[由方程(10.2)和(10.3)确定的多步预估校正方法，并非全部可用矩阵符号来表示．但是，若对预估系数加以限制，这是可能的]我们只讨论能够这样表示的对一般 p 阶方程的方法，因此这些方法可变换成标准形式．对于特殊 p 阶方程的多步方法，常常能写成这种形式，然而如果校正公式 (10.2) 精确地被求解，用多步的符号来处理通常更方便．

10.1. 收敛性和稳定性

收敛性表达了用充分小的步长和准确的计算使数值解可任意接近真实解的性质，即欲知在区域 $0 < t \leqslant b$ 中固定点 t 上的解，可以选择一个充分大的 N，使得如果 $h = \frac{t}{N}$，则 y_N 按我们的需要充分接近于 $y(t)$． 在多值或多步方法中，从初始向量 $\mathbf{a}_0$ 或 $\mathbf{y}_0$ 开始，但初始向量可能没有（由初始值）完全确定．例如对于一阶方程的二步方法，我们给出了 y_0，但还需要知道 y_1．由于用到的任何数值方法几乎一定要在 y_1 中（也

可能在 y_0 中)引进误差，所以，如果要实际可行的话，我们必须在收敛性定义中涉及到这些误差．从而有下述定义．

定义 10.1. 一阶方程的多步(多值)方法称为收敛的，如果对任何满足 Lipschitz 条件的微分方程，当 $\mathbf{y}_0 \to \mathbf{y}(0)[\mathbf{a}_0 \to \mathbf{a}(0)]$ 和 $h = \frac{t}{n}(n \to \infty)$ 时，计算得到的解 $\mathbf{y}_n[\mathbf{a}_n]$ 在区间 $0 \leqslant t \leqslant b$ 上一致收敛于 $\mathbf{y}(t)[\mathbf{a}(t)]$．

如果我们求 p 阶方程的解，为了求解，有 $y_{(0)}, y'_{(0)}, \cdots, y_{(0)}^{(p-1)}$ 的 p 个初始值，但是方法仍可能需要附加的起始值．我们必须要求所用的起始值当 $h \to 0$ 时能正确表示初始值；即 $\mathbf{a}_0$ 的分量都满足

$$q!\,\frac{(\mathbf{a}_0)_q}{h^q} \to y^{(q)}(0) \qquad \text{对 } q < p,$$

$$\frac{(\mathbf{a}_0)_q}{h^{p-1}} \to 0 \qquad \text{对 } q \geqslant p,$$

其中 $(\mathbf{a})_q$ 为 $\mathbf{a}$ 的第 q 个分量．

我们用按下式

$$\|\mathbf{a}\|_h^* = \max_i \frac{|(\mathbf{a})_i|}{h^{q(i)}}$$

定义的模 $\|\cdot\|_h^*$ 来处理它们，其中

$$q(i) = \begin{cases} i & \text{如果 } i < p, \\ p-1 & \text{如果 } i \geqslant p. \end{cases}$$

必须要求对 p 阶方法的起始值 $\mathbf{a}_0$ 按这个模收敛于 $\mathbf{a}(0)$，即

$$\|\mathbf{a}_0 - \mathbf{a}(0)\|_h^* \to 0 \qquad \text{当 } h \to 0.$$

象下面的例子可以看到的那样，较大的误差相当于初始条件的改变．

例.

考虑由 $\mathbf{a}_n = [y_n, hy'_n, h^2y''_n/2]^T$

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{I} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

给出的对二阶方程 $0 = f(y, y', t) - y''$ 的方法，它与

$$0 = \frac{h^2}{2} f\left(a_0, \frac{a_1}{h}, t\right) - a_2 \tag{10.8}$$

是等价的. 如果取特殊情形 $f(y, y', t) = 0$，我们得到

$$\mathbf{a}_N = S\mathbf{a}_{N-1}, \tag{10.9}$$

其中

$$S = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \end{bmatrix}.$$

(10.9) 的解为 $\mathbf{a}_N = S^N \mathbf{a}_0$，其中

$$S^N = \begin{bmatrix} 1 & N & 2N-1 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \end{bmatrix}.$$

因此，当起始向量为 $[y_0, hy'_0, h^2 y''_0/2]^T$ 时，数值解为

$$y_N = y_0 + Nhy'_0 + (2N-1)\frac{h^2 y''_0}{2}.$$

由于 $Nh = t$，有

$$y_N = y_0 + ty'_0 + (2t - h)\frac{hy''_0}{2}.$$

分量 $(\mathbf{a}_0)_1$ 的大小为 ε 的一个误差，使解改变成

$$y_N = y_0 + t\left(y'_0 + \frac{\varepsilon}{h}\right) + (2t - h)\frac{hy''_0}{2}.$$

所以，为了收敛性，显然 ε 必须按 $o(h)$ 变化. $h^2 y''_0/2$ 中的初始误差也必须按 $o(h)$ 变化. 而 y_0 的误差只要象 $o(1)$ 一样收敛于零.

虽然我们要求起始值按模 $\|\cdot\|_h^*$ 给出的方式收敛于初始值，但我们仅关心出现在 f 中的 $\mathbf{a}_n$ 的那些分量. 因此，有下述

定义.

定义 10.2. p 阶方程的多值方法称为 q 收敛的，如果对满足 Lipschitz 条件的任意 p 阶 q 微分方程，当$\|\mathbf{a}_0-\mathbf{a}(0)\|_h^q \to 0$ 和 $n\to\infty$，$h=\frac{t}{n}$ 时，计算解 $\mathbf{a}_n$ 使

$$\|\mathbf{a}_n-\mathbf{a}(t)\|_h^{p-q+1}\to 0$$

对 $0\leqslant t\leqslant b$ 是一致的.

我们简单地称 1-收敛方法为收敛的方法

10.1.1.稳定性

在第 1 章中，如下定义稳定性，初值上的一个小扰动，当 h 减小到零时，只在解上引起一个有界的变化.由于单步方法中的每一步实际上是一个新的初值问题，因此，稳定性用来限制计算中由于每一步的小扰动引起的变化. 在多步方法中，希望有一个类似的要求，但是我们必须体现对于 p 阶方程解的一些变化可能相当于在等价初值问题中引起大的变化. 对于一阶方程，可以不变地采用定义，但是在导出方程(10.9)的例子中看到，当 $h\to 0$ 时，hy_0' 中一个固定的改变量引起了 y_N 的一个任意大的变化，因此，我们采用下述的稳定性定义.

定义 10.3. 一个多步(多值)方法对于一阶方程是稳定的，如果对于任何满足 Lipschitz 条件的一阶方程，存在常数 k 和 h_0，使对所有 $0\leqslant t\leqslant b$ 和所有 $h=\left(\frac{t}{n}\right)\in(0,h_0)$，关系式

$$\|\mathbf{y}_n-\mathbf{y}_n^*\|\leqslant k\|\mathbf{y}_0-\mathbf{y}_0^*\|,$$
$$[\|\mathbf{a}_n-\mathbf{a}_n^*\|\leqslant k\|\mathbf{a}_0-\mathbf{a}_0^*\|] \qquad (10.10)$$

均成立，其中 $\mathbf{y}_n$ 和 $\mathbf{y}_n^*$ [$\mathbf{a}_n$ 和 $\mathbf{a}_n^*$] 是两个数值解.

我们将看到，对于 p 阶方程还需要一种形式的稳定性，其中起始值的扰动只在微分方程中出现的导数上引起有界的变

化．因此，我们定义：

定义 10.4. 一个多值方法对 p 阶方程是 q 稳定的，如果对于任何满足 Lipschitz 条件的 p 阶 q 微分方程，存在常数 K 和 h_0，使

$$\|\mathbf{a}_n-\mathbf{a}_n^*\|_h^{p-q+1}\leqslant k\|\mathbf{a}_0-\mathbf{a}_0^*\|_h^p \tag{10.11}$$

对所有 $0\leqslant t\leqslant b$ 和所有 $h=\left(\frac{t}{n}\right)\in(0,\ h_0)$ 均成立。

上面给出的稳定性定义是合乎要求的，但不是一个方便的定义．我们需要一个对微小的变化不太灵敏的方法，可是需要一个检验稳定性的实际可行的方法，这就是根条件．

定义 10.5. 一阶方程的多步方法满足根条件，如果 $\rho(\xi)=0$ 的所有根都在单位圆内，或者在单位圆上（是单根）．一阶方程的多值方法满足根条件，如果 $S=(I-\mathbf{l}\delta_1^T)^M A$ 的所有特征值均在单位圆内，并且对应于线性初等因子[1]．

对 p 阶方程，定义 q 根条件．它对应于 q 稳定性．

定义 10.6. p 阶方程的多值方法满足 q 根条件，如果 $S=(I-\mathbf{l}\delta_p^T)^M A$ 的特征值都在单位圆内，或在单位圆上．单位圆上的特征值不对应于秩大于 p 的初等因子．如果单位圆上有一个秩为 p 的特征值，则单位圆上无其它特征值有大于 q 的秩．

我们将看到，相容性要求特征值 1 具有秩 p，所以，这条

1）如果矩阵 S 用相似变换归化成它的 Jordan 标准形，重复的特征值在对角线上产生形如

$$\begin{bmatrix}\xi_i & 1 & & 0\\ & \ddots & \ddots & \\ & & \ddots & 1\\ 0 & & & \xi_i\end{bmatrix}$$

的 $m\times m$ 块．这是一个秩为 m 的初等因子．如果 m 为 1，它是线性初等因子．

件实际上限制了附加的特征值的秩，使其不超过 q.我们称 1 根条件为严格根条件.

下面两个定理表明了根条件的必要性.

定理 10.1. 如果一阶方程的多步方法是稳定的，则多项式 $\rho(\xi)$ 满足根条件.

定理 10.2. 如果多值方法是 q 稳定的，它必须满足 q 根条件.

在证明这些结果时，两个引理是有用的，并且在后面将是需要的.

引理 10.1. 差分方程

$$-y_n + \alpha_1 y_{n-1} + \cdots + \alpha_k y_{n-k} = 0 \quad \alpha_k \neq 0 \quad (10.12)$$

的解可以表示成

$$y_n = \sum_{i=1}^{s} \sum_{j=1}^{m_i} C_{ij} n^{j-1} \xi_i^n, \qquad (10.13)$$

其中 ξ_i 为

$$\rho(\xi) = -\xi^k + \sum_{i=1}^{k} \alpha_i \xi^{k-i} = 0$$

的 s 个不同的根，而且 ξ_i 的重数为 m_i. C_{ij} 由 $y_i(0 \leqslant i < k)$ 的初始条件唯一确定.

证明. 将 (10.13) 代入 (10.12)的左端，并令 $\alpha_0 = -1$，我们得到

$$\sum_{i=1}^{s} \sum_{j=1}^{m_i} C_{ij} \sum_{l=0}^{k} \alpha_l \xi_i^{n-l} (n-l)^{j-1}$$

$$= \sum_{i=1}^{s} \sum_{j=1}^{m_i} C_{ij} \left(\left(\xi \frac{d}{d\xi} \right)^{j-1} (\xi^{n-k} \rho(\xi)) \right)_{\xi=\xi_i}.$$

如果 ξ_i 是 ρ 的 m_i 重根，则

$$\xi^{n-k} \rho(\xi) = (\xi - \xi_i)^{m_i} Q(\xi),$$

其中 $Q(\xi)$ 为 ξ 的多项式. 因此

$$\left(\left(\xi\frac{d}{d\xi}\right)^{j-1}(\xi^{n-k}\rho(\xi))\right)_{\xi=\xi_i}=0\quad(j\leqslant m_i).$$

于是由(10.13)给出的 y_n 为(10.12)的解. 当 $y_l(0\leqslant l<k)$ 给定时,方程(10.12)唯一确定 y_n, $(n\geqslant k)$, 所以,仅需证明这些 y_l 唯一确定 C_{ij}. 这由下面的事实推出, 即对于 $n=0$, $1,\cdots,k-1$, (10.13)是以 y_l 表示的 C_{ij} 的非奇异线性方程组. [Henrici (1962) 给出方程组的行列式值

$$\prod_{1\leqslant\mu<\nu\leqslant s}(\xi_\mu-\xi_\nu)^{m_\mu+m_\nu}\prod_{\nu=1}^{s}(m_\nu-1)!!,$$

其中 $0!!=1$ 和 $k!!=k!((k-1)!!)$. 这个行列式不为零.]

引理 10.2. 如果矩阵 S 有特征值 ξ_i, 而 ξ_i 对应于秩 m^i 的初等因子集,则

1. 如果所有 $|\xi_i|<1$, 则当 $n\to\infty$ 时, S^n 的所有元素 $\to 0$;

2. 如果任意一个 $|\xi_i|>1$, 则当 $n\to\infty$ 时, S^n 中有元素 $\geqslant O(\xi_i^n)$[1)]

3. 如果所有 $|\xi_i|\leqslant 1$, 并且使得 $|\xi_i|=1$ 的最大 m_i 为 m, 则当 $n\to\infty$ 时, S^n 中有元 $=O(n^{m-1})$, 但是没有更大的.

证明. 我们将 S 表示成它的 Jordan 型

$$T^{-1}ST=\Delta=\begin{bmatrix}\xi_1 & 1 & & & & & \\ & \ddots & 1 & & & & \\ & & \xi_1 & & & & \\ & & & \xi_2 & 1 & & \\ & & & & \ddots & 1 & \\ & & & & & \xi_2 & \\ & & & & & & \ddots\ \xi_s\end{bmatrix}$$

---

1) 记号 $a(x)\geqslant O(x)$ 用来表示存在常数 k, 使 $|a(x)/kx|\geqslant 1$ 对它的极限值的一个邻域中的所有 x 均成立. 在现在的情形,它表示 S^n 的有些元素至少以 $|\xi_i^n|$ 那样快的速度趋向无穷大.

因此

$$S^n = (T\Delta T^{-1})^n = T\Delta^n T^{-1}.$$

具有特征值 ξ 且阶为 m 的初等因子将导致 Δ^n 中的对角线块

$$\begin{bmatrix} \xi^n & n\xi^{n-1} & \cdots & \binom{n}{m-1}\xi^{n-m+1} \\ & \ddots & \ddots & \\ & & & n\xi^{n-1} \\ 0 & & & \xi^n \end{bmatrix}$$

1. 如果 $|\xi|<1$，这块的所有元素→0，因此，如果所有 $|\xi_i|<1$，则 Δ^n 的所有元素→0.

2. 如果 $|\xi_i|>1$，则对某些分量有 $S^n\mathbf{x}=\xi_i^n\mathbf{x}=O(\xi_i^n)$，其中 $\mathbf{x}$ 为对应于 ξ_i 的特征向量. 因此，S^n 中一些分量 $\geqslant O(\xi_i^n)$.

3. 如果 $|\xi_i|=1$，则对应于这个 ξ_i 的 Δ^n 块，其右上角元素是 $O(n^{m_i-1})$. 由于 T^{-1} 是非奇异的，因此，存在 $\mathbf{x}$ 使 $T^{-1}\mathbf{x}=[0,\cdots,0,1,0,\cdots,0]^T$，其中 1 出现在对应于这一块的最右边元素的位置上. 于是

$$S^n\mathbf{x} = T\Delta^n T^{-1}\mathbf{x} = T\begin{bmatrix} & 0 & \\ \hline & \begin{matrix}\xi_i^n & \binom{n}{m_i-1}\xi_i^{n-m+1} \\ & \ddots \\ & \xi_i^n\end{matrix} & \\ \hline & 0 & \end{bmatrix}\begin{bmatrix}0\\ \vdots \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0\end{bmatrix}$$

$$= T\begin{bmatrix}0\\ \vdots \\ 0 \\ \binom{n}{m_i-1}\xi_i^{n-m_i+1} \\ \vdots \\ \xi_i^n \\ 0 \\ \vdots \\ 0\end{bmatrix}$$

它具有量级为 $O(n^{m_i-1})$ 的元素. 没有向量 $\mathbf{x}$ 能使 $S^n\mathbf{x}$ 大于 $O(n^{m-1})$.

定理 10.1 的证明: 如果由于 $\rho(\xi)$ 的一个根 ξ 有 $|\xi|>1$ 使根条件不满足,则方程 $y'=0$ 当 $\mathbf{y}_0=0$ 时的解给出 $\mathbf{y}_N=0$, 又如果 $\mathbf{y}_0^*$ 使起始值对应于解 $y_i^*=\xi^i(0\leqslant i<k)$, 则当 $N\to\infty$ 时, $\mathbf{y}_N^*$ 是无界的. 如果 ξ 是模为 1 的根, 且重数 $m>1$, 则由 $y_i^*=i^{m-1}\xi^i(0\leqslant i<k)$ 的起始值 $\mathbf{y}_0^*$ 导致解

$$|\mathbf{y}_N|=|N^{m-1}\xi^N|$$

当 $N\to\infty$ 时无界. 这样, 在这两种情形中, $\|\mathbf{y}_0-\mathbf{y}_0^*\|$ 是有界的, 但 $\|\mathbf{y}_N-\mathbf{y}_N^*\|$ 是无界的, 所以, 对多步方法, 由于根条件的破坏, 也破坏了稳定性.

定理 10.2 的证明: 对具有

$$\mathbf{a}_0=[0,\ \cdots,\ 0]^T$$

的问题

$$y^{(p)}=0$$

考虑满足

$$\|\delta\mathbf{a}_0\|_h^p=O(1)$$

的起始值中的偏差

$$\delta\mathbf{a}_0=\mathbf{a}_0^*-\mathbf{a}_0.$$

解为

$$\mathbf{a}_N^*=S^N\delta\mathbf{a}_0.$$

由引理 10.2, 如果 S 的一个特征值 ξ 大于 1, 则 S^N 的一些分量 $\geqslant O(\xi^N)$; 因此, $\mathbf{a}_N^*$ 的一些分量是无界的. 如果模为 1 的特征值 ξ 有秩 m, 则 S^N 的一些分量是 $O(N^{m-1})$. 因此, $\mathbf{a}_N^*$ 的一些分量为 $O(N^{m-1}h^{p-1})=O(N^{m-p})$. 当 $m>p$ 时, 它们均无界.

由于对 q 稳定性 $\|\mathbf{a}_n^*\|_h^{p-q+1}$ 必须为 $O(1)$, (在最大的意义下) S^n 按 n 的幂次可能会有"最坏的"变化, 表示在下面的表达式中:

$$
\mathbf{a}_n^* = O\left(\begin{bmatrix}
1\\
n^{-1}\\
\cdots\\
n^{q-p}\\
\hdashline
n^{q-p}\\
\cdots\\
n^{q-p}\\
\hdashline
n^{q-p}\\
\cdots\\
n^{q-p}
\end{bmatrix}\right) =
$$

$$
O\left(\overset{S^n}{\left[\begin{array}{cccc:ccc:ccc}
1 & n & \cdots & n^{p-q} & n^{p-q+1} & \cdots & n^{p-1} & n^{p-1} & \cdots & n^{p-1}\\
 & 1 & \cdots & n^{p-q-1} & n^{p-q} & \cdots & n^{p-2} & n^{p-2} & \cdots & n^{p-2}\\
 & 0 & \cdots & & \cdots & & \cdots & \cdots & & \cdots\\
 & & & 1 & n & \cdots & n^{q-1} & n^{q-1} & \cdots & n^{q-1}\\
\hdashline
 & & & 1 & n & \cdots & n^{q-1} & n^{q-1} & \cdots & n^{q-1}\\
 & 0 & & \cdots & \cdots & & \cdots & \cdots & & \cdots\\
 & & & 1 & n & \cdots & n^{q-1} & n^{q-1} & \cdots & n^{q-1}\\
\hdashline
 & & & 1 & n & \cdots & n^{q-1} & n^{q-1} & \cdots & n^{q-1}\\
 & 0 & & \cdots & \cdots & & \cdots & \cdots & & \cdots\\
 & & & 1 & n & \cdots & n^{q-1} & n^{q-1} & \cdots & n^{q-1}
\end{array}\right]}
\overset{\delta\mathbf{a}_0}{\begin{bmatrix}
1\\
n^{-1}\\
\cdots\\
n^{q-p}\\
\hdashline
n^{q-1-p}\\
\cdots\\
n^{1-p}\\
\hdashline
n^{1-p}\\
\cdots\\
n^{1-p}
\end{bmatrix}}\right)
$$

如果在单位圆上 S 有两个或更多个特征值，其中一个秩为 p，而另外一个的秩大于 q，则 q 稳定性被破坏[1]．证完．

将定理 10.2 的证明作点简单的推广，得到

定理 10.3. q 收敛的多值方法满足 q 根条件．

由此推出，特殊 p 阶方程的多步方法 (10.2) 在单位圆上

1) 假定在单位圆上至少有两个特征值，其秩 p 和 $\tilde{q}$ 满足 $p \geq \tilde{q} > q$，即可看出．如果将 S 写成 $T\Delta T^{-1}$，并且只考虑 Δ^n 中大小为 $n^{\tilde{q}-1}$ 的分量，我们可以在 S 中找到对应的分量．由于 $\tilde{q} > q$，为了保持上面的形式，它们只可能出现在 0 到 $p-\tilde{q}$ 行中．在 Δ^n 中恰好有 $p-\tilde{q}+2$ 个量级为 $n^{\tilde{q}-1}$ 的分量．它们出现在不同的行和列．它们占有了 $p-\tilde{q}+2$ 个 T 的列和 T^{-1} 的行．由于结果中仅有 $p-\tilde{q}+1$ 行非零，因此，或者 T 的那些列或者 T^{-1} 的那些行必须线性相关．但 T 是非奇的，因此 $\tilde{q} \ngtr q$．

不能有超过 p 重的根，因为它与多值方法是等价的.

10.1.2.阶

对一阶方程，多步方法的阶在定义 8.1 中已经定义.这个定义容易推广到由(10.2)给出的特殊 p 阶方程 $y^{(p)}=f(y,t)$ 的多步方法.

定义 10.7. 如果算子 L_h 由

$$L_h(y(t))=\sum_{i=0}^{k}\left(\alpha_i y(t-hi)+\frac{h^p}{p!}\beta_i y^{(p)}(t-hi)\right)$$

定义，于是阶 r 是使所有 $y\in C_{r+1}$ 均有

$$L_h(y(t))=O(h^{r+1})$$

的最大整数.

如果 $y\in C_{r+2}$，我们可以用带 $O(h^{r+2})$ 余项的 Taylor 级数代替 $y(t-hi)$ 和 $h^q y^{(q)}(t-hi)$，得到

$$L_h(y(t))=\sum_{q=0}^{r+1}C_q h^q y^{(q)}(t)+O(h^{r+2}),\qquad(10.14)$$

其中

$$C_q=\begin{cases}\displaystyle\sum_{i=0}^{k}\frac{(-i)^q}{q!}\alpha_i & (q<p),\\ \displaystyle\sum_{i=0}^{k}\left[\frac{(-i)^q}{q!}\alpha_i+\frac{(-i)^{q-p}}{p!(q-p)!}\beta_i\right] & (r+1\geqslant q\geqslant p).\end{cases}\qquad(10.15)$$

这表明阶由方法的系数确定. 如果象以前一样定义多项式 ρ 和 σ，我们看到

$$C_q=\begin{cases}\displaystyle\frac{1}{q!}\left[\left(\xi\frac{d}{d\xi}\right)^q(\xi^{-k}\rho(\xi))\right]_{\xi=1} & (q<p),\\ \displaystyle\frac{1}{q!}\left[\left(\xi\frac{d}{d\xi}\right)^q(\xi^{-k}\rho(\xi))\right. & \\ \displaystyle\qquad\left.+\binom{q}{p}\left(\xi\frac{d}{d\xi}\right)^{q-p}(\xi^{-k}\sigma(\xi))\right]_{\xi=1} & (q\geqslant p).\end{cases}\qquad(10.16)$$

由此推出，如果阶 $r \geqslant p$，则

$$\begin{aligned} \rho(1) &= 0, \\ \rho'(1) &= 0, \\ &\cdots\cdots \\ \rho^{(p-1)}(1) &= 0, \\ \rho^{(p)}(1) + \sigma(1) &= 0. \end{aligned} \tag{10.17}$$

反之，如果(10.17)成立，由于当 $q \leqslant p$ 时 $C_q = 0$，阶 $\geqslant p$. 注意这仅根据校正公式确定方法的阶，即方法中的公式是显式，或者校正一直迭代到收敛的情形.

显然，多值方法或者 PC 多步方法的阶必须用微分方程的解来定义.

定义 10.8. 如果 $\mathbf{a}(t)$ 是对某个 h 在时刻 t 的向量 $\mathbf{a}$ 的正确值，并且定义

$$\begin{aligned} \tilde{\mathbf{a}}_{(0)} &= A\mathbf{a}(t-h), \\ \tilde{\mathbf{a}}_{(m+1)} &= \tilde{\mathbf{a}}_{(m)} + \mathbf{l}F(\tilde{\mathbf{a}}_{(m)}), \\ \tilde{\mathbf{a}}(t) &= \tilde{\mathbf{a}}_{(M)}. \end{aligned} \tag{10.18}$$

于是 p 阶方程的方法的阶是使得若 F 表示具有解 $y \in C_{r+1}$ 的任意 p 阶微分方程，则

$$\tilde{\mathbf{a}}(t) - \mathbf{a}(t) = O(h^{r+1}) \tag{10.19}$$

成立的最大的 r.

对一阶方程，已知预估公式的阶可以小于校正公式的阶，而且每次校正迭代结果的阶都增加 1，直到达到校正公式的阶为止. 类似地，对特殊的 p 阶方程，可以用多步显式预估公式，并且对每次校正迭代，它的阶将增加 p，直到校正的阶为止.

多值方法的阶依赖于求积的方程. 例如，预估方程可能是 q 阶，而校正方程的阶 $r > p + q$. 于是单个的校正迭代公式可以得到阶为 $q + 1$的方法. 但是，对于微分方程 $y^{(p)} =$

$f(t)$，由于没有用预估公式，方法的阶为 r. 这样，定义 10.8 给出的阶是在所有具有光滑解的方程上的最小数. 如果限制方程的类别(例如限制成 q 微分方程)，则方法的阶更高. 显然，对给定类方程的方法的阶，由方法的系数唯一确定. 但是，我们还没有用这些系数的代数关系式来表示阶. 这将推迟到 10.4 节因为确定阶的上极限是复杂的，而且实际的格式不太重要，在这一节，我们导出关于系数的一些必要条件，它们在收敛性证明中是有用的.

如果预估-校正格式可以表示成多值方法的格式［如果 $p=1$，或者 f 仅依赖于 y 和 t，或者关于 $y', \cdots, y^{(p-1)}$ 预估公式的限制都满足，就可以做到这一点］，则当把方法写成标准形式后，预估的阶容易从矩阵 A 的形式中看出. 如果预估的阶为 r'，而校正的阶至少为 r'，则 A 的前 $r'+1$ 列形如

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ & 1 & \cdots & r' \\ & & \ddots & r' \\ & 0 & & 1 \\ \hline & & 0 & \end{bmatrix}$$

即它们在上部组成 Pascal 三角形矩阵，而下部为零.

引理 10.3. 如果 p 阶方程标准形式的多值方法有阶 $\geqslant p-1$，则

$$S=(I-\mathbf{l}\delta_p^T)^M A$$

的前 p 列在上部组成 Pascal 三角形矩阵，而在下部为零. 如果预估公式的阶小于 $p-1$，则 $l_p=1$.

证明. 考虑问题

$$y^{(p)}=p!.$$

令 $\mathbf{a}(t)$ 的第 i 个分量为 $t^{p-i}h^i\binom{p}{i}$. 对这个问题，有

$$F(\mathbf{a}) = h^p - a_p,$$

其中 a_p 为 $\mathbf{a}$ 的第 p 个分量. 因此, $\tilde{\mathbf{a}}_{(m+1)} = \tilde{\mathbf{a}}_{(m)} + \mathbf{l}F(\tilde{\mathbf{a}}_{(m)})$ 的第 p 个分量为 $(1 - l_p)(\tilde{\mathbf{a}}_{(m)})_p + l_p h^p$, 使得

$$F(\tilde{\mathbf{a}}_{(m+1)}) = (1 - l_p)F(\tilde{\mathbf{a}}_{(m)}) = (1 - l_p)^{m+1}F(\tilde{\mathbf{a}}_{(0)}).$$

因此,

$$\tilde{\mathbf{a}}_{(M)} = \tilde{\mathbf{a}}_{(0)} + \mathbf{l}\sum_{i=0}^{M-1}(1 - l_p)^i F(\tilde{\mathbf{a}}_{(0)}).$$

为了避免混乱,如果 $l_p \neq 0$, 则记 $\mathbf{l}(1 - (1 - l_p)^M)/l_p$ 为 $\tilde{\mathbf{l}}$; 如果 l_p 为零,记 $\tilde{\mathbf{l}}$ 为 $\mathbf{l}M$. 因此

$$\begin{aligned}\tilde{\mathbf{a}}(t) = \tilde{\mathbf{a}}_{(M)} &= \tilde{\mathbf{a}}_{(0)} + \tilde{\mathbf{l}}F(\mathbf{a}_{(0)}) \\ &= \tilde{\mathbf{l}}h^p + (I - \tilde{\mathbf{l}}\boldsymbol{\delta}_p^T)A\mathbf{a}(t - h).\end{aligned}$$

如果阶至少为 $p - 1$, $\tilde{\mathbf{a}}(t)$ 与 $\mathbf{a}(t)$ 最多可以相差 $O(h^p)$. 因此, $(I - \mathbf{l}\boldsymbol{\delta}_p^T)A$ 的前 p 列必须是所述形式,因为 $\mathbf{a}(t - h)$ 的前 p 个元素都比 $O(h^p)$ 大. 容易看出

$$S = (I - \mathbf{l}\boldsymbol{\delta}_p^T)^M A = (I - \tilde{\mathbf{l}}\boldsymbol{\delta}_p^T)A.$$

因此,结果的第一部分由此获得.

如果预估的阶$<p - 1$,则 A 的某一行在第 p 个元素的左面含有非 Pascal 三角形矩阵的元素. 用 $(I - \mathbf{l}\boldsymbol{\delta}_p^T)$ 左乘 A 的结果,是从第 i 行减去第 p 行的 l_i 倍. 如果 $l_p = 1$,在第一次这样的一步后,第 p 行为零, 但是如果 $l_p \neq 1$, 则任何有限次迭代后, 它仍是非零的. 因此, 如果第 p 行对角线元素的左边(即 0 列到 $p - 1$ 列)有非零元素,那么,为了它们在 S 中为零,有 $l_p = 1$. 如果在任意行的第 p 列左端有非 Pascal 三角形元素, 则在第 p 行的同样列中必须有非零元素.

这个结果容易推广成

引理 10.4. 如果 p 阶方程标准形式的多值方法有阶 $r \leq k - 1$, 则 $(I - \mathbf{l}\boldsymbol{\delta}_p^T)^M A$ 的前 $r + 1$ 列与 $(I - \mathbf{l}\boldsymbol{\delta}_p^T)^M \tilde{A}$ 的前 $r + 1$ 列一致, 其中 $\tilde{A}$ 是 Pascal 三角形矩阵. 证明留给读者,

即问题 4.

正如下面的定理所表明的，阶除了提供误差变化的性态外，对一个方法来说，极小阶还是必要条件.

定理 10.4. 如果多步方法（10.2）对 p 阶方程是收敛的，则（10.2）的阶至少为 p.

证明. 我们用方程（10.17）作为 p 阶方法的一个表征. 注意到 $\xi=1$ 为 $\rho(\xi)=0$ 的 $j+1$ 重根，其充要条件为它是 $\rho(\xi)$ 的 i 重根，而且有

$$\sum_{i=0}^{k}\alpha_i p_j(m+i)=0, \tag{10.20}$$

其中 p_j 为任意 i 次多项式[1]. 考虑方程 $y^{(p)}=0$，其初始值除 $y^{(j)}(0)=j!(j<p)$ 外为 $y^{(q)}(0)=0, 0\leqslant q<p$. 这个问题的解是 $y=t^j$. 我们考察从正确的起始值 $y_i=z_ih^j$ 出发的（10.2）的解，其中 $z_i=i^j$. 对此我们得到解 $y_n=z_nh^j$，这里

$$\sum_{i=0}^{k}\alpha_i z_{n-i}=0. \tag{10.21}$$

如果方法是收敛的，对固定的 t_n，$y_n/t_n^j\to 1$，所以

1）可以这样来证明：如果 $\xi=1$ 是 $\rho(\xi)$ 的 $j+1$ 重根，则对 $0\leqslant q\leqslant j$ 有

$$0=\left(\frac{d^q}{d\xi^q}(\rho(\xi)\xi^n)\right)_{\xi=1}=\sum_{i=0}^{k}\alpha_i\tau_q(i-n-k),$$

其中 $\tau_q(x)=-x(-x-1)\cdots(-x-q+1)$ 为 x 的 q 次多项式. 由于 $\tau_q(x)(0\leqslant q\leqslant j)$ 是 j 次多项式空间的基底，可将任意 j 次多项式 $p_j(x)$ 表成 $\sum_{q=0}^{j}\nu_q\tau_q(x)$. 如果令 $m=-n-k$，立即得到

$$\sum_{i=0}^{k}\alpha_i p_j(m+i)=\sum_{q=0}^{j}\nu_q\sum_{i=0}^{k}\alpha_i\tau_q(i-n-k)=0.$$

反之，如果（10.20）成立，它对 $p_j(m+i)=\tau_j(i-k)$ 成立，所以

$$\left(\frac{d^j}{d\xi^j}\rho(\xi)\right)_{\xi=1}=0.$$

如果 $\xi=1$ 是 $\rho(\xi)=0$ 的 j 重根，则可推出它是 $j+1$ 重根.

$$\frac{z_n}{n^j} \to 1, \qquad 当\ n \to \infty. \tag{10.22}$$

对于 $j = 0$，(10.21) 的极限推出

$$\sum_{i=0}^{k} \alpha_i = 0.$$

由此推出：由于 (10.20)，$\xi = 1$ 是 $\rho(\xi)$ 的根. 现在假定 $\xi = 1$ 为 ρ 的 j 重根，$j \geqslant 1$，由 (10.21)，有

$$\begin{aligned}
0 &= \sum_{n=k}^{N} p_{j-1}(2k - n) \sum_{i=0}^{k} \alpha_i z_{n-i} \\
&= p_{j-1}(k)[\alpha_0 z_N + \alpha_1 z_{N-1} + \cdots + \alpha_k z_{N-k}] \\
&\quad + p_{j-1}(k-1)[\alpha_0 z_{N-1} + \cdots + \alpha_{k-1} z_{N-k} \\
&\quad + \alpha_k z_{N-k-1}] + \cdots + p_{j-1}(2k - N + 1) \\
&\quad \times [\cdots + \alpha_k z_1] + p_{j-1}(2k - N)[\cdots \\
&\quad + \alpha_{k-1} z_1 + \alpha_k z_0] \\
&= \sum_{n=N-k}^{N} z_n \sum_{i=0}^{N-n} \alpha_i p_{j-1}(n - N + k + i) \\
&\quad + \sum_{n=k}^{N-k-1} z_n \sum_{i=0}^{k} \alpha_i p_{j-1}(n - N + k + i) \\
&\quad + \sum_{n=0}^{k-1} z_n \sum_{i=k-n}^{k} \alpha_i p_{j-1}(n - N + k + i).
\end{aligned}$$

由 (10.20)，第二项为零，而第三项为 N 的 $j-1$ 次多项式. 除以 N^j，当 $N \to \infty$ 时最后项 $\to 0$，而对 $n \in [N-k, N]$，$z_n/N^j \to 1$. 因此，得到

$$\begin{aligned}
0 &= \sum_{n=N-k}^{N} \sum_{i=0}^{N-n} \alpha_i p_{j-1}(n - N + k + i) \\
&= \sum_{i=0}^{k} \alpha_i \sum_{n=N-k+i}^{N} p_{j-1}(n - N + k + i) \\
&= \sum_{i=0}^{k} \alpha_i \sum_{n=0}^{k-i} p_{j-1}(n + 2i).
\end{aligned}$$

由于 $p_{j-1}(n)$ 是 n 的 $j-1$ 次多项式，$\sum_{n=0}^{k-i} p_{j-1}(n+2i)$是 i 的 j 次多项式，因此，从 (10.20)，$\xi=1$ 是 $\rho(\xi)$ 的 $(j+1)$ 重根. 由于我们只限制 $j<p$，推出 ξ 是 $\rho(\xi)$ 的 p 重根，而且

$$\rho(1)=\rho'(1)=\cdots=\rho^{(p-1)}(1)=0.$$

为了完成这个证明，必须证明满足 (10.17) 的最后一个方程. 为此，考虑问题 $y^{(p)}=p!, y^{(q)}(0)=0, 0\leqslant q<p$. 其解是 $y=t^p$. 差分方程 (10.2) 给出

$$\sum_{i=0}^{k}\alpha_i y_{n-i}+h^p\sum_{i=0}^{k}\beta_i=0. \tag{10.23}$$

由于 $\xi=1$ 是 $\rho(\xi)$ 的 p 重根，

$$\sum_{i=0}^{k}\alpha_i(n-i)^p=\left[\left(\xi\frac{d}{d\xi}\right)^p(\xi^{n-k}\rho(\xi))\right]_{\xi=1}=\rho^{(p)}(1).$$

由于方法是收敛的，定理 10.3 告诉我们，$\rho^{(p)}(1)\neq 0$，因此

$$y_n=-(hn)^p\frac{\sigma(1)}{\rho^{(p)}(1)} \tag{10.24}$$

满足 (10.23). $y_i(0\leqslant i<k)$ 的值与正确起始值 $(hi)^p$ 之间的差最多为 $O(h^p)$，所以收敛的方法将使 $y_n\to t_n^p=(hn)^p$. 这和 (10.24) 推出 (10.17) 最后的方程，所以收敛的方法有阶 $\geqslant p$. 这就是所要证明的.

虽然猜想一般预估-校正多步或多值方法的阶也必须 $\geqslant p$，但是，除了 $p=1$ 多步方法的情形外，还尚未证明.

定理 10.5. 对一阶方程的预估-校正方法，如果它是收敛的，其阶必 $\geqslant 1$.

证明. 若确定预估和校正公式的多项式分别为 ρ, σ 和 ρ^*, σ^*. 由于定理 10.4 证明了校正公式的阶至少为 1，所以只要考虑预估的阶为 -1 和阶为 1 或更大的单个校正的情形. 进行规格化使 $\alpha_0=\alpha_0^*=-1$，由方程 (10.18) 和 (10.19) 给出的阶的定义，有

$$\sum_{i=1}^{k}[\alpha_i^* y(t-ih)+h\beta_i^* y'(t-ih)]$$

$$+h\beta_0^* f\left[\sum_{i=1}^{k}[\alpha_i y(t-ih)+h\beta_i y'(t-ih)]\right]$$

$$-y(t)=O(h^{r+1}) \tag{10.25}$$

对任意解 $y(t)\in C_{r+1}$ 的微分方程 $y'=f(y, t)$ 均成立. 我们考虑方程 $y'=y$. 对此方程,由 (10.25) 得到

$$e^{t-kh}\{\rho^*(e^h)+h\sigma^*(e^h)+h\beta_0^*[\rho(e^h)+h\sigma(e^h)]\}=O(h^{r+1}). \tag{10.26}$$

如果 ξ 是

$$\rho^*(\xi)+h\sigma^*(\xi)+h\beta_0^*\rho(\xi)+h^2\beta_0^*\sigma(\xi)=0 \tag{10.27}$$

的根,则 $y_n=\xi^n$ 为预估-校正方法的解. 对形如 $\xi=e^h+\Delta$ 的解,我们考察 (10.27),其中 Δ 是个小量. 代入后得到

$$\Delta\rho^{*\prime}(e^h)+\rho^*(e^h)+h\sigma^*(e^h)+h\beta_0^*\rho(e^h)+h^2\beta_0^*\sigma(e^h)=O(\Delta^2+h\Delta).$$

由 (10.26) 和 $\rho^{*\prime}(e^h)=\rho^{*\prime}(1)+O(h)$, 左端的最后四项为 $O(h^{r+1})$, 所以

$$\Delta=\frac{1}{\rho^{*\prime}(1)}O(\Delta^2+h\Delta+h^{r+1}).$$

由定理 10.4, 校正公式有阶 1 ,因此 $\rho^*(1)=0$. 由于方法收敛,它满足根条件 (定理 10.3), 所以 $\rho^*(\xi)$ 在 $\xi=1$ 无重根. 因此有 $\rho^{*\prime}(1)\neq 0$.

于是

$$\Delta=kh^{r+1}+O(h^{r+2}), \qquad k\neq 0.$$

如果 $r=0$, 对 $t=nh$ 有

$$y_n=\xi^n=(e^h+kh+O(h^2))^n=e^{(k+1)t}+O(h^2).$$

因此, 如果预估公式的阶为 -1, 则初值 $y(0)=1$, 起始值

$y_i=(e^h+\Delta)^i(0\leqslant i<k)$ 的初值问题 $y'=y$ 的解收敛于与解 e^t 不同的解．证完．

10.1.3. 相容性和收敛性

定义 10.9. p 阶方程的方法称为相容，如果它的阶至少为 p.

一个大概正确的结果为具有 q 稳定性和相容性是一般方程（10.1）q 收敛的必要充分条件．

我们已经证明了这个结果的一部分，即

1. q 稳定性 $\rightarrow q$ 根条件．
2. q 收敛性 $\rightarrow q$ 根条件．
3. 收敛性→只用校正公式的方法的相容性．
4. 收敛性→对一阶方程 PC 方法的相容性．

这一节我们证明：

5. q 根条件和相容性 $\Longrightarrow q$ 收敛性．
6. q 根条件和相容性 $\Longrightarrow q$ 稳定性．

我们猜想有

7. 收敛性→对 p 阶方程多值方法的相容性．

我们从最简单的情形，即一阶的单个方程开始证明收敛性

定理 10.6. 一阶方程的稳定相容的多值方法是收敛的．

证明．虽然可以对 $f(y,t)$ 仅满足连续性和 Lipschitz 条件证明这个结果，但需要假定 f 有连续的一阶导数（除有限个点外，通常如此，而这有限个点，容易处理）．在这种情形，$y\in C_2$，并且将量 $\tilde{\mathbf{a}}_{(m)}$ 和 $\tilde{\mathbf{a}}(t)$ 关于 $t-h$ 的 Taylor 级数代入由（10.18）和（10.19）给出的阶的定义中，得到

$$\tilde{\mathbf{a}}(t)-\mathbf{a}(t)=P(y,y',y'',f,f_y,f_t,h)h^2,$$

其中 $P(\cdots)$ 是它的变量多项式，并且变量值在差分解与微分

方程解之间的某个点上取值[1]．由于变量可限制在解的范围上，存在 D，使得

$$\|\tilde{\mathbf{a}}(t_n)-\mathbf{a}(t_n)\|=\|\mathbf{d}_n\|\leqslant Dh^2 \tag{10.28}$$

T 是导数的界的组合．

在方程 (9.12) 和 (9.13) 中，我们已经证明

$$\mathbf{e}_n=S_n\mathbf{e}_{n-1}+\mathbf{d}_n, \tag{10.29}$$

其中

$$S_n=\prod_{i=0}^{M-1}\left[I+\mathbf{l}\frac{\partial F}{\partial \mathbf{a}}(\xi_i)\right]A.$$

对于标准形式的多值方法，

$$\frac{\partial F}{\partial \mathbf{a}}=[hf_y(\xi_i),\ -1,\ 0,\ \cdots,0]=hf_y(\xi_i)\boldsymbol{\delta}_0^T-\boldsymbol{\delta}_1^T.$$

由于 $|f_y|<L$，我们可以写出

$$S_n=S+h\tilde{S}_n,$$

其中

$$S=[I-\mathbf{l}\boldsymbol{\delta}_1^T]^MA,$$

而 $\tilde{S}_n$ 为矩阵[2]，这个矩阵的元素是具有有界系数的 h 的多项式．

---

1) 如果为了达到收敛允许进行任意次校正迭代，则命题必须改变，因为多项式可有任意高的阶，并且不能用简单地限制它的变量来求界．在这种情形，注意有

$$\tilde{\mathbf{a}}(t)=\mathbf{a}_{(0)}+\omega\mathbf{l},$$

其中 ω 表示所有用来达到收敛的校正的总量．因此，$\tilde{\mathbf{a}}(t)$ 满足 $F(\tilde{\mathbf{a}}(t))=(\tilde{\mathbf{a}}(t))_1-hf((\tilde{\mathbf{a}}(t))_0)=0$．因而

$$(\tilde{\mathbf{a}}_{(0)})_1+\omega l_1-hf((\tilde{\mathbf{a}}_{(0)})_0+\omega l_0)=0.$$

如果 L 为 f_y 的上界，只要 $h<h_0=l_1/(l_0L)$，这个方程对 ω 有唯一解．这个解由

$$\omega=\frac{1}{l_1}[hf((\tilde{\mathbf{a}}_{(0)})_0)-(\tilde{\mathbf{a}}_{(0)})_1][1-hl_0(f_y/l_1)]^{-1}$$

给出，其中 f_y 在适当的点上取值．如果对固定的 $\tilde{h}<h_0$，保持 $h\leqslant\tilde{h}$，则可记

$$\tilde{\mathbf{a}}(t)=\mathbf{a}_{(0)}+\frac{1}{l_1}[hf((\tilde{\mathbf{a}}_{(0)})_0)-(\tilde{\mathbf{a}}_{(0)})_1][1+hl_0(f_y/l_1)+kh^2],$$

其中 k 是有界的．于是 $\tilde{\mathbf{a}}(t)-\mathbf{a}(t)$ 可表成具有界系数的多项式．

2) 如果允许任意次校正迭代，则必须应用脚注 D 中所用的同样技巧，来得到对 $\tilde{S}_n$ 的具有有界系数的多项式．

因此，

$$\|\tilde{S}_n\| \leqslant C_0 \qquad \text{对 } h \leqslant h_0,$$

并且从 (10.29) 导出

$$\|\mathbf{e}_n\| = \left\| \sum_{j=1}^{n} S^{n-j}(h\tilde{S}_j\mathbf{e}_{j-1} + \mathbf{d}_j) + S^n\mathbf{e}_0 \right\|$$

$$\leqslant \sum_{j=1}^{n} \|S^{n-j}\|(hC_0\|\mathbf{e}_{j-1}\| + \|\mathbf{d}_j\|) + \|S^n\|\|\mathbf{e}_0\|. \tag{10.30}$$

由关于 S 的稳定性条件推出 $\|S^m\| \leqslant C_1$，它与 m 无关. 现在用通常的方法可以找到 (10.30) 的一个上界. 将 $\|S^m\|$ 和 $\|d_j\|$ 用它们的界代替，我们得到

$$\|\mathbf{e}_N\| + \frac{Dh}{C_0} \leqslant \sum_{j=1}^{n} hC_0C_1\left(\|\mathbf{e}_{j-1}\| + \frac{Dh}{C_0}\right) + \frac{Dh}{C_0} + C_1\|\mathbf{e}_0\|. \tag{10.31}$$

取 (10.31) 中的等号，并将其看成对 $\|\mathbf{e}_n\| + (Dh/C_0)$的递推关系式，我们找到形如

$$\|\mathbf{e}_N\| + \frac{Dh}{C_0} = K(1 + hC_0C_1)^N, \qquad N \geqslant 1$$

的解，并且找到

$$K = \frac{Dh}{C_0} + C_1\|\mathbf{e}_0\|.$$

于是

$$\|\mathbf{e}_N\| \leqslant \left[\frac{Dh}{C_0} + C_1\|\mathbf{e}_0\|\right](1 + hC_0C_1)^N - \frac{Dh}{C_0}. \tag{10.32}$$

由于 $(1 + hC_0C_1)^N \leqslant e^{hNC_0C_1} \leqslant e^{bC_0C_1} = \tilde{K}$，这里 $\tilde{K}$ 与 h 无关，我们可用

$$\|\mathbf{e}_N\| \leqslant K_1h + K_2\|\mathbf{e}_0\| \tag{10.33}$$

来得到误差的界. 于是，如果当 $h \to 0$ 时 $\|\mathbf{e}_0\| \to 0$，则对所有使 $Nh \leqslant b$ 的N都有 $\|\mathbf{e}_N\| \to 0$. 这就是所要证明的.

(10.33)提供的界是粗糙的，并且难以应用. C_0与$\tilde{S}_n$的大小无关，它由 Lipschitz 常数确定，但是对一般的方法计算C_0是困难的. 定理 10.6 的证明可直接用来证明下列结果:

定理 10.7. 如果一阶方程的多值方法有阶 r，并且起始误差 $\|\mathbf{e}_0\|$以$D'h^{r'}$为界,解 $y(t)' \in C_{r+1}$, 则在时刻 $t = Nh$ 的误差界于

$$\|\mathbf{e}_N\| \leqslant \frac{Dh^r}{C_0}(e^{C_0C_1t}-1) + D'h^{r'}C_1e^{C_0C_1t}. \quad (10.34)$$

由于 $y \in C_{r+1}$,局部截断误差以 Dh^{r+1} 为界. 将这个界代替上面证明中的 Dh^2, 即得结果.

由定理 10.7 提供的界，虽然是右渐近阶的形式，但它仍受到取模时丢掉许多信息的影响. 这个界可以稍微改进，但进一步的改进必须在得到如同单步方法中作出估计时才行.

现在给出对微分方程组

$$(y^i)^{(p_i)} = f^i(\{y^j\}, \{y'^j\}, \cdots, t), \; i, j = 1, \cdots, s \quad (10.35)$$

的方法的一般收敛性和误差界的定理，方程组中每个方程可以有不同的阶 p_i，并且认为出现在右边的第 j 个因变量不会超过 $p_j - q_j$ 阶的导数,其中 $1 \leqslant q_j \leqslant p_j$. 第 i 个方程用方法

$$\begin{aligned} \mathbf{a}^i_{n,(0)} &= A_i\mathbf{a}^i_{n-1}, \\ \mathbf{a}_{n,(m+1)} &= \mathbf{a}^i_{n,(m)} + \mathbf{I}_iF^i(\{\mathbf{a}^j_{n,(m)}\}), \quad (10.36) \\ \mathbf{a}^i_n &= \mathbf{a}^i_{n,(M)} \end{aligned}$$

来处理，其中 $\mathbf{a}^i$ 为第 i 个变量的标准形式多值方法的信息，而

$$F^i(\{\mathbf{a}^j\}) = \frac{h^{p_i}}{p!}f^i\left(\{a^j_0\}, \left\{\frac{a^j_1}{h}\right\}, \left\{2!\frac{a^j_2}{h^2}\right\}, \cdots t\right) - a^i_p, \quad (10.37)$$

其中 a^i_k 是 $\mathbf{a}^i$ 的第 k 个分量.

定理 10.8. 如果由 A_i, $\mathbf{I}_i$ 给出的方法满足 p_i 阶方程的 q_i 根条件,且它们的阶$r_i \geqslant p_i$ (即它们对 q_i 微分方程是相容的).如果 f^i 是充分连续可微的，则倘若第 i 个变量在起始值中的误

差 $\|\mathbf{a}_0^i-\mathbf{a}^i(0)\|_h^{p_i}$以$D'h^{r'_i}$ 为界和第 i 个方程是 q_i 微分的，就有

$$\|\mathbf{a}_n^i-\mathbf{a}^i(t_n)\|_h^{p_i-q_i+1}=O(h^d),$$

其中

$$d=\min(r'_i, r_i-p_i+1). \tag{10.38}$$

这些方法可以变换成标准形式．假定已经做了这个变换．

证明．在这个证明中，分下列几步：

1. 证明第 i 个方程的局部截断误差以 $D^ih^{p_i+d}$ 为界．
2. 求 S^n 的元素的界．
3. 重复定理 10.6 的证明．

第一步

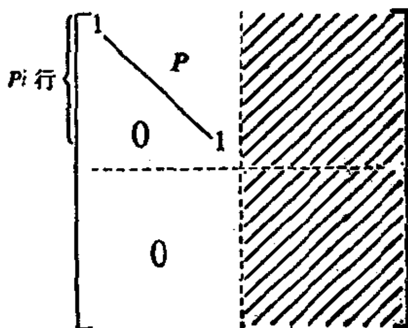
由于对 p_i 阶方程由 $A_i, \mathbf{l}_i$ 给出的应用M次校正迭代方法，有阶 r_i，预估公式有阶 $\geqslant r_i-M\geqslant p_i+d-M$，于是，预估步的截断误差以 $D^ih^{p_i+d-M}$ 为界（通过用 Taylor 级数求得）．在校正步，每个 $\mathbf{a}^j$ 通过项 $h^{p_i}f^i(\cdots, y^j, \cdots(y^j)^{(p_j-1)}, \cdots)$ 能参加到其它 $\mathbf{a}^i$ 的计算中，所以可能出现的最坏的误差由项 $h^{p_i}(y^j)^{(p_j-1)}=h^{p_i}a^j_{p_j-1}(p_j-1)!/h^{p_j-1}$产生．在这些项中，误差为 $O(h^{p_i}h^{p_j+d-M}/h^{p_j-1})$ 或 $O(h^{p_i+d-M+1})$．利用带余项的 Taylor 级数，对 $h\leqslant h_0$ 可以作成形如 $D^ih^{p_i+d-M+1}$ 的界．我们看到，校正一次阶的界就增加 1．重复 $M-1$ 次，得到需要的界．

第二步

由于引理 10.3 和方法 A_i, $\mathbf{l}_i$ 是相容的

$$S_i=(I-\mathbf{l}_i\boldsymbol{\delta}_{p_i}^T)^MA_i$$

有形式



其中阴影部分表示可能的非零元素，P 是 Pascal 三角形矩阵，所有元素均非零. 因此，左上角是特征值为 1 和阶为 p_i 的单个初等因子块. 由于 S_i 满足 q_i 根条件，所以其它特征值都小于 1 或等于 1，但具有秩 $\leqslant q_i$ 的初等因子. 于是，对 n 用归纳法容易看到 S_i^n 的元素的量级为 n 的幂次

$$
\left[\begin{array}{cccccc:ccc}
1 & n & n^2 & \cdots & & n^{p_i-1} & n^{p_i-1} & \cdots & n^{p_i-1} \\
 & 1 & n & \cdots & & n^{p_i-2} & n^{p_i-2} & \cdots & n^{p_i-2} \\
 & & \ddots & & & \cdots & \cdots & & \cdots \\
 & 0 & & \ddots & & n^{q_i-1} & n^{q_i-1} & \cdots & n^{q_i-1} \\
 & & & & \ddots & \cdots & \cdots & & \cdots \\
 & & & & & n & & & \\
 & & & & & 1 & n^{q_i-1} & \cdots & n^{q_i-1} \\
\hdashline
 & & & 0 & & & & n^{q_i-1} &
\end{array}\right]
$$

第三步

定义 $\mathbf{e}_n^i = \mathbf{a}_n^i - \mathbf{a}^i(t_n)$ 和 $\mathbf{e}_{n,(m)}^i = \mathbf{a}_{n,(m)}^i - \tilde{\mathbf{a}}_{(m)}^i(t_n)$，有

$$\mathbf{e}_{n,(0)}^i = A_i\mathbf{a}_{n-1}^i - A_i\mathbf{a}^i(t_n - h) = A_i\mathbf{e}_{n-1}^i, \qquad (10.39)$$

$$
\begin{aligned}
\mathbf{e}_{n,(m+1)}^i &= \mathbf{a}_{n,(m)}^i + \mathbf{l}_i F^i(\{\mathbf{a}_{n,(m)}^i\}) - \tilde{\mathbf{a}}_{(m)}^i(t_n) \\
&\quad - \mathbf{l}_i F^i(\{\tilde{\mathbf{a}}_{(m)}^i(t_n)\}) \\
&= \mathbf{e}_{n,(m)}^i + \mathbf{l}_i \sum_j \frac{\partial F^i}{\partial \mathbf{a}^i} \mathbf{e}_{n,(m)}^j,
\end{aligned} \qquad (10.40)
$$

其中 $\partial F^i/\partial \mathbf{a}^j$ 在正确解和数值解之间某个点上求值，求和的上下限是明显的，已省掉.

$$\mathbf{e}_n^i = \mathbf{a}_{n,(M)}^i - \mathbf{a}^i(t_n) = \mathbf{e}_{n,(M)}^i + \mathbf{d}_n^i, \qquad (10.41)$$

其中 $\mathbf{d}_n^i$ 为第 i 个方程在第 n 步的截断误差. 现在

$$\frac{\partial F^i}{\partial \mathbf{a}_j} = -\delta_j^i \boldsymbol{\delta}_{p_i}^T + \sum_{q=0}^{p_j-1} \frac{h^{p_i-q}}{p_i!} q! \frac{\partial f^i}{\partial y^{j(q)}} \boldsymbol{\delta}_q^T,$$

其中，如果 $i \neq j$，$\delta_j^i = 0$；如果 $i = j$，$\delta_j^i = 1$. 将其代入

(10.40)，则得到

$$\mathbf{e}^i_{n,(m+1)} = (I - \mathbf{I}_i\boldsymbol{\delta}^T_{pi})\mathbf{e}^i_{n,(m)} + \sum_j \sum_q h^{p_i-q}\frac{q!}{p_i!}\frac{\partial f^i}{\partial y^{j(q)}}\mathbf{I}_i\boldsymbol{\delta}^T_q\mathbf{e}^j_{n,(m)}. \tag{10.42}$$

因此，从(10.41)，(10.42)和(10.39)，

$$\mathbf{e}^i_n = S_i\mathbf{e}^i_{n-1} + \mathbf{d}^i_n + \sum_{m=0}^{M-1}(I - \mathbf{I}_i\boldsymbol{\delta}^T_{pi})^{M-1-m}\sum_j\sum_q h^{p_i-q}\frac{q!}{p_i!}\frac{\partial f^i}{\partial y^{j(q)}}\mathbf{I}_i\boldsymbol{\delta}^T_q\mathbf{e}^j_{n,(m)}$$
$$= S_i\mathbf{e}^i_{n-1} + \mathbf{d}^i_n + U^i_n, \tag{10.43}$$

其中三重求和项已记成U^i_n(注意，出现在U^i_n中的f^i的偏导数通常对不同的m是在不同的点上求值的.这对问题无影响，因为我们只需要这些项的粗糙的界)．重复应用(10.43)，得到

$$\mathbf{e}^i_N = \sum_{n=1}^{N} S_i^{N-n}(\mathbf{d}^i_n + U^i_n) + S_i^N\mathbf{e}_0. \tag{10.44}$$

我们对$N \geqslant 1$用归纳法证明对某些常数k，k_1，k_2，$k_{1,m}$，$k_{2,m}$有

$$\|\mathbf{e}^i_N\|^{s_i}_h \leqslant h^d[(1+hk)^N(k_1+k_2) - k_2] \tag{10.45}$$

和

$$\|\mathbf{e}^i_{n,(m)}\|^{s_i}_h \leqslant h^d[(1+hk)^{N-1}(k_{1,m}+k_{2,m}) - k_{2,m}], \tag{10.46}$$

其中$s_i = p_i - q_i + 1$．这就完成了证明．

如果选$k_1 \geqslant D'$，对$N=0$不等式(10.45)成立，我们对N，然后对$m \leqslant N$用二重归纳进行．假定(10.45)对所有$0 \leqslant N \leqslant n-1$成立，由(10.39)，$k_{1,0}$和$k_{2,0}$可用$k_1$和$k_2$来表示，使(10.46)对$m=0$，$1 \leqslant N \leqslant n$是满足的． 现在假定(10.46)对于$\mathbf{e}^i_{N,(m)}$，所有$1 \leqslant N \leqslant n$，$0 \leqslant m \leqslant m'$成立．由(10.46)对$q < s_j$有

$$|\boldsymbol{\delta}^T_q\mathbf{e}^j_{n,(m)}| \leqslant h^{d+q}[(1+hk)^{n-1}(k_{1,m}+k_{2,m}) - k_{2,m}], \tag{10.47}$$

所以，利用所出现的偏导数的界，方程(10.42)说明，可以选取$k_{1,m+1}$和$k_{2,m+1}$，使(10.46)对$m = m'+1$成立． 因此，由对

$0 \leqslant N \leqslant n-1$ 的(10.45)推出(10.46)对 $0 \leqslant N \leqslant n$ 成立. 将(10.46)代到(10.43)中 U_n^i 的定义里,并应用(10.47),我们得到对某些常数 k_{1U} 和 k_{2U},有

$$\|U_n^i\| \leqslant h^{p_i+d}[(1+hk)^{n-1}(k_{1U}+k_{2U})-k_{2U}].$$

而从第一步我们有

$$\|\mathbf{d}_n^i\| \leqslant Dh^{p_i+d}.$$

令 ϵ 是分量均为 1 的列向量. 由第二步中找到的 S_i^m 的形式,有

$$\|S_i^n \epsilon\|_h^{s_i} \leqslant k_4 h^{-p_i+1} \qquad \text{对 } nh \leqslant b. \tag{10.48}$$

因此,从(10.44)得到

$$\|\mathbf{e}_N^i\|n_h^{s_i} \leqslant h^d k_4\left[\sum_{n=1}^{N} h(D+(1+hk)^{n-1}(k_{1U}+k_{2U})-k_{2U})+D'\right].$$

将 k_{2U} 用 $k_5=\max(D, k_{2U})$ 代替,有

$$\|\mathbf{e}_N^i\|_h^{s_i} \leqslant h^d k_4\left[\frac{(1+hk)^N-1}{k}(k_{1U}+k_5)+D'\right]. \tag{10.49}$$

注意,由(10.48),k_4 是固定的且不依赖于其它 k 的值;特别,$k_4 \geqslant 1$,所以可设 $k_1=k_4D'$. 如果 k_2 已选定,k_5 和 k_{1U} 被确定,于是可选 k 使

$$\frac{k_4}{k}(k_{1U}+k_5)=k_1+k_2.$$

将这些量代入到(10.49),就归结成(10.45),这证明了方法按 $O(h^d)$ 收敛. 证完.

如果在定理 10.8 中我们认为 $\mathbf{e}_n$ 是单个 p 阶方程以不同的起始值 $\mathbf{a}_0$ 和 $\mathbf{a}_0^*$ 开始的两个数值解 $\mathbf{a}_n^*$ 和 $\mathbf{a}_n$ 的差,则可应用第二步和第三步,只要令其中的 $\mathbf{d}_n$ 为零. 因此,$D=0$.在整个证明中,我们可选 $k_2=k_{2,m}=k_{2U}=k_5=0$. 注意 $k_1=k_4D'$,所以若设 $\|\mathbf{a}_0^*-\mathbf{a}_0\|_h^s=D'$,则有 $r'=0,d=0$ 以及由(10.45)推出

$$\|\mathbf{a}_N^* - \mathbf{a}_N\|_h^s = \|\mathbf{e}_N\|_h^s \leqslant k_4(1 + hk)^N D'$$
$$\leqslant K\|\mathbf{a}_0^* - \mathbf{a}_0\|_h^s, \qquad 对\ 0 \leqslant hN \leqslant b,$$

其中 k 与 $\|\mathbf{a}_0^* - \mathbf{a}_0\|_h^s$ 无关. 令 $s = p - q + 1$, 我们就证明了如下定理.

定理 10.9 如果相容的多值方法满足根条件, 则它是 q 稳定的.

10.2. 稳定多步方法的最高阶

我们现在回头讨论 Dahlquist 的基本结果, 它给出了对一阶方程稳定多步方法最高阶的一个界. 由前述知, 可以选取 ρ 和 σ, 使阶达到 $2k$. 现在证明稳定性的重要要求将限制阶为 $k+1$(若 k 是奇数)和 $k+2$(若 k 为偶数). 已知对任何 ρ 可以选取 σ 使得阶为$k+1$.由于可以选 ρ 是稳定的,所以容易达到 $k+1$ 阶. 我们将看到,当且仅当 k 是偶数以及 ρ 选取成使它的所有根均在单位圆上, 才能选取 σ 使得阶为 $k+2$.

稳定性与 $\rho(\xi)$ 的根均在单位圆内或圆上的条件是等价的. 为了便于讨论,作变换使得

$$\xi = \frac{1+z}{1-z} \text{ 或 } z = \frac{\xi - 1}{\xi + 1}.$$

这个变换将单位圆的内部映射到左半平面, 这由 $\xi = re^{i\theta} = rc + irs$ 可以看出, 其中 $c = \cos\theta$ 和 $s = \sin\theta$. 我们有

$$z = \frac{rc + irs - 1}{rc + irs + 1} = \frac{[(rc+1) - irs][(rc-1) + irs]}{[(rc+1) - irs][(rc+1) + irs]}$$
$$= \frac{(r^2 - 1) + 2irs}{(rc+1)^2 + r^2s^2}.$$

分母大于零;如果 $r^2 < 1$,分子在左半平面;如果 $r^2 = 1$,分子在虚轴上;如果 $r^2 > 1$, 分子在右半平面.因此, 图 10.2 中阴影区互相对应.

我们定义

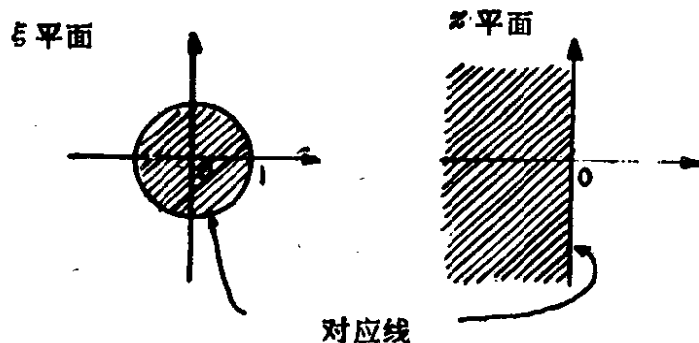


图 10.2

$$R(z)=\left(\frac{1-z}{2}\right)^{k}\rho\left(\frac{1+z}{1-z}\right)$$

和

$$S(z)=\left(\frac{1-z}{2}\right)^{k}\sigma\left(\frac{1+z}{1-z}\right).$$

它们都是 z 的 k 次多项式. $\rho(\xi)=0$ 的根(使 $z\neq 1$ 的根)对应于 $R(z)=0$ 的根. 假定 ρ 是稳定的,则使 $z=1$ 的 ξ 不是 $\rho(\xi)=0$ 的根. 于是,除了 ρ 在-1 的根外, R 和 ρ 的所有根均相对应. ρ 在 -1 的根对应于 R 在无穷远的根, 即要降低 R 的次数. 假定方法的阶为 r. 证明按下列步骤进行:

代入 ξ 的变换式并乘以 $((1-z)/2)^{k}$, 将方法为 r 阶的条件[见方程 (8.10)],即

$$\frac{\rho(\xi)}{\log(\xi)}+\sigma(\xi)=O((\xi-1)^{r})$$

映成

$$\frac{R(z)}{\log\{(1+z)/(1-z)\}}+S(z)=O(z^{r}).$$

将 $R(z)/\log\{(1+z)/(1-z)\}$ 展成 z 的幂级数,如 $\sum_{n=0}^{\infty}r_{n}z^{n}$, 并按 k 为奇数或偶数对 $n=k+1$ 或$k+2$ 证明 $r_{n}<0$. 由于 $S(z)$ 为最大次数 k 的多项式我们看出, 当 k 是奇数或偶

数时有 $r \leqslant k+1$ 或 $k+2$.

首先，我们考虑 $R(z)=a_0+a_1z+\cdots+a_kz^k$. 由于 $\xi=1$ 是 $\rho(\xi)=0$ 的单根，$z=0$ 是 $R(z)=0$ 的单根，于是 $a_0=0, a_1 \neq 0$. 不失一般性，可取 $a_1>0$. 我们证明，稳定性 $\Longrightarrow a_i \geqslant 0 (i \geqslant 2)$. ρ 是实多项式，因之 R 也是实的. 因此，R 的根或者是实根 x_μ，或者按共轭对 $x_\nu \pm iy_\nu$ 出现，其中 x_μ 和 $x_\nu \leqslant 0$. 由于在 ξ 平面上 ρ 在单位圆内的根映成 R 在 z 左半平面上的根可将 $R(z)$ 写成

$$
\begin{aligned}
& a \prod_\mu (z-x_\mu) \prod_\nu (z-x_\nu-iy_\nu)(z-x_\nu+iy_\nu) \\
& = a \prod_\mu (z+|x_\mu|) \prod_\nu (z^2+2z|x_\nu|+x_\nu^2+y_\nu^2).
\end{aligned}
\tag{10.50}
$$

于是，每个系数都具有 a 的符号，但 $a_1>0$，因此，$a_i \geqslant 0, (i \geqslant 2)$. 下面考虑

$$
\begin{aligned}
& \frac{z}{\log[(1+z)/(1-z)]} \\
& = \sum_{\mu=0}^{\infty} c_{2\mu} z^{2\mu} \\
& = \frac{z}{\log(1+z)-\log(1-z)} \\
& = \frac{z}{z-(z^2/2)+(z^3/3)-\cdots+z+(z^2/2)+(z^3/3)+\cdots} \\
& = \frac{1}{2+(2z^2/3)+(2z^4/5)+\cdots}.
\end{aligned}
$$

于是有 $c_0=\frac{1}{2}$. 我们证明 $c_{2\mu}<0\ (\mu \geqslant 1)$. 这是下面 Kaluza (1928)引理的特殊情形.

引理 10.5. 如果

$$\left(\sum_{n=0}^{\infty} a_n x^n\right)\left(\sum_{n=0}^{\infty} b_n x^n\right) = 1,$$

$a_n > 0, n \geqslant 0$ 和 $a_{n-1} \cdot a_{n+1} > a_n^2,\ n \geqslant 1$，则对 $n \geqslant 1$ 有 $b_n < 0$.

证明. 不失一般性，令 $a_0 = 1$. 于是 $b_0 = 1$. 如果考察含 x^n 的项，则有

$$0 = a_n + \sum_{k=1}^{n} a_{n-k} b_k. \tag{10.51}$$

通过观察含 x^{n+1} 的项，我们得到

$$b_{n+1} = -a_{n+1} - \sum_{k=1}^{n} a_{n+1-k} b_k. \tag{10.52}$$

将 (10.51) 乘上 a_{n+1}, (10.52) 乘上 a_n，然后相加，得到

$$a_n b_{n+1} = \sum_{k=1}^{n} b_k (a_{n+1} a_{n-k} - a_n a_{n+1-k}). \tag{10.53}$$

我们按归纳法进行. 如果 $b_1, \cdots, b_n < 0$，由于 $a_n > 0$ 和 $a_{n+1} a_{n-k} - a_n a_{n+1-k} > 0$，(10.53) 证明有 $b_{n+1} < 0$ (这可以从考虑 $a_{n+1}\, a_n\, a_{n-1}^2 \cdots a_{n-k+2}^2\, a_{n-k+1}\, a_{n-k} = (a_{n+1}\, a_{n-1})\,(a_n\, a_{n-2}) \times (a_{n-1}\, a_{n-3}) \cdots (a_{n-k+2}\, a_{n-k}) > a_n^2\, a_{n-1}^2 \cdots a_{n-k+1}^2$ 并除以 $a_n a_{n-1}^2 \cdots a_{n-k+2}^2 a_{n-k+1}$ 看出). 由于 $a_0 b_1 + b_0 a_1 = b_1 + a_1 = 0$, $b_1 = -a_1 < 0$，所以结果对 $n = 1$ 是正确的. 证完.

在引理 10.4 中取 $z^2 = x$ 和 $a_n = 2/(2n+1)$，由于 $a_n > 0$ 和 $a_{n-1} a_{n+1} = 4/(2n-1)(2n+3) = 4/[(2n+1)^2 - 4] \geqslant a_n^2$，立即得到对 $\mu \geqslant 1$ 有 $c_{2\mu} = b_\mu < 0$.

回到我们原来的问题，现在我们考虑

$$\frac{R(z)}{z} \frac{z}{\log[(1+z)/(1-z)]} = \sum_{m=1}^{k} a_m z^{m-1} \sum_{\mu=0}^{\infty} c_{2\mu} z^{2\mu}.$$

由于这是 $\sum_{n=0}^{\infty} r_n z^n$，所以有

$$r_{k+1} = c_2 a_k + c_4 a_{k-2} + \cdots + c_{k+1} a_1, \text{ 对奇数 } k \quad (10.54)$$

和

$$r_{k+1} = c_2 a_k + c_4 a_{k-2} + \cdots + c_k a_2, \text{ 对偶数 } k. \quad (10.55)$$

由于 $a_1 > 0$，$a_j \geq 0$ 和 $c_{2\mu} < 0$，(10.54) 表示对奇数 k 有 $r_{k+1} < 0$，这证明了 $k+1$ 是最高的稳定阶．如果 k 是偶数，(10.55)只当 $a_2 = a_4 = \cdots = a_k = 0$ 时能给出 $r_{k+1} = 0$．在这种情形，

$$r_{k+2} = c_4 a_{k-1} + c_6 a_{k-3} + \cdots + c_{k+2} a_1 < 0,$$

所以最高阶为 $k+2$，而且通过取 $r_{k+1}=0$，它是能够达到的．由(10.55)，需要 $a_2 = a_4 = \cdots = 0$．由于 a_0 也是零，$R(z)$ 为 z 的奇多项式．因此，$R(-z) = -R(z)$．$R(z)$ 的根也是 $R(-z)$ 的根，所以，若 $x_\nu + iy_\nu$ 是 $R(z)$ 的根，则 $-x_\nu - iy_\nu$ 亦是．但是，由于稳定性，两个根都应该在左半平面中，从而推出 $x_\nu = 0$，所以 $R(z)$ 的所有根都在虚轴上，表示 $\rho(\xi)$ 的根都在单位圆上．反之，如果 $\rho(\xi)$的根均在单位圆上，则 $R(z)$的根都在虚轴上，从而推出 $R(z)$是奇的，而且 $a_2 = a_4 = \cdots = 0$，所以，若适当选取多项式 $\sigma(\xi)$，阶为 $k+2$．这样就证明了下面的定理．

定理10.10. 一阶方程稳定 k 步方法的最高阶为

$k+1$，如果 k 是奇数；

$k+2$，如果 k 是偶数．

$k+2$ 的阶只能出现在 $\rho(\xi)$ 的根均在单位圆上的情形．

$k+2$ 的阶推出方法是弱稳定的，所以这样的方法限制了实用上的重要性．

对高阶方程的方法，已经得到类似的结果[例如，Dahlquist (1959)]．但是，由于下一节多值方法的结果，这些结果实际

效果极微.

10.3. 稳定多值方法的存在性

这一节证明存在稳定的最高阶 k 值方法. 最高的阶到底是多少,下一节进一步讨论. 但是,我们已经看到,$k-1$ 步 Adams-Bashforth-Moulton 预估-校正方法是 k 阶的,且等价于 k 值方法,这表示对一阶方程其阶可以是 k.

在 p 阶方程的方法中,必须保证

$$S=(I-\mathbf{l}\delta_p^T)^M A$$

满足稳定性条件. S 的 p 个特征值是 1(主根). 我们证明可以选取 $\mathbf{l}$, 使最高阶方法的其余特征值能取任意需要的值.用 $\mathbf{a}$ 中具有 k 个值的标准形式来表示,如果预估公式最高阶为 $k-1$, 则 A 是 Pascal 三角形矩阵.

定理 10.11. 如果 A 是 $k\times k$ Pascal 矩阵, $A_{ij}=\binom{i}{j}$ $(0\leqslant i,j\leqslant k-1)$, 则对任何 $k-p$ 个数 $\{\lambda_i\}$ 的组, 存在列向量 $\mathbf{l}$,使矩阵

$$(I-\mathbf{l}\delta_p^T)^M A$$

有 p 个特征值等于 1 和 $k-p$ 个特征值等于组 $\{\lambda_i\}$, 这特征值与 $\mathbf{l}$ 的前 p 个元素无关, 并唯一确定 $\mathbf{l}$ 的后 $k-p$ 个元素.

证明. $\mathbf{l}$ 的前 p 个元素的无关性从矩阵的形式来看是显然的. 这个形式也推出前 p 个特征值为 1 ,于是可取 $p=0$, A 为 A 的 $(k-p)\times(k-p)$ 的下主子式. A 所需要的唯一性质是它具有相等的非零对角线元素和非零第一非对角线元素 $A_{i,i+1}$ 的上三角形. 对于 Pascal 矩阵的所有主子式,这是成立的. 首先注意

$$(I-\mathbf{l}\delta_0^T)^M=I-c\mathbf{l}\delta_0^T,$$

其中

$$c = \begin{cases} -\dfrac{(1-l_0)^M - 1}{l_0}, & l_0 \neq 0, \\ M & l_0 = 0. \end{cases}$$

因此，只要考虑 $M = 1$．令

$$T = A(I - \mathbf{l}\delta_0^T).$$

由于 A 是非奇异的并且 $A^{-1}TA = S$，故有与

$$S = (I - \mathbf{l}\delta_0^T)A$$

同样的特征值．$T = A - \mathbf{u}\delta_0^T$，其中 $\mathbf{u} = A\mathbf{l}$．在下面的引理中，我们证明存在 A 的相似变换 Q，将 A 变成在对角线紧上面具有非零元素的 Jordan 型，而且其中 Q 和 Q^{-1} 有三角形的形式

$$\begin{bmatrix} d_0 & 0 & 0 & \cdots & 0 \\ 0 & d_1 & x & \cdots & x \\ 0 & 0 & d_2 & \cdots & x \\ & \cdots & & \ddots & \cdot \\ 0 & 0 & 0 & \cdots & d_{k-1} \end{bmatrix}$$

其中 x 表示非零元素．

于是得到

$$Q^{-1}TQ = Q^{-1}AQ - Q^{-1}\mathbf{u}\delta_0^TQ = Q^{-1}AQ - \mathbf{v}\delta_0^T,$$

其中 $\mathbf{v} = Q^{-1}\mathbf{u}d_0$．除对角线上的 1 外，$Q^{-1}TQ$ 是系数为 $\mathbf{v}$ 的元素的多项式的伴随矩阵形式．因此，可以选择 $\mathbf{v}$ 使根等于 $\{\lambda_i\}-1$．在这种情形，$\mathbf{l} = A^{-1}Q\mathbf{v}/d_0$ 使 S 具有所需要的特征值．这就唯一确定了 $\mathbf{l}$．

尚须证明下面引理的正确性．

引理10.6. 如果 A 是上三角形矩阵，使 $A_{ii} = c \neq 0$（i 不是求和）以及如果 $A_{i,i+1} \neq 0$（$i = 0, 1, \cdots, k-1$），于是存在具有上述形式的相似变换 Q，将 A 变换成它的对角线元素不变的 Jordan 型．

证明．将 Q_i 构造成如下的初等变换：

$$Q_i = \begin{bmatrix} 1 & 0 & & & \cdots & 0 \\ 0 & 1 & & & \cdots & 0 \\ \cdots & & & & \cdots & \\ 0 & \cdots & 1 & q_{i+1,i+2} & \cdots & q_{i+1,k-1} \\ \cdots & & & & \cdots & \\ 0 & 0 & & & \cdots & 1 \end{bmatrix}$$

并令 $A_{i+1} = Q_i^{-1}A_iQ_i(i = 0, 1, \cdots, k-3)$，其中 $A_0 = A$. 可以选取 $q_{i+1,j}$ 使 A_{i+1} 的第 i 行最后 $k-i-2$ 个元素为零，同时不改变对角线或第一非对角线元素.因此，$\tilde{Q} = Q_0Q_1\cdots Q_{k-3}$ 使 A_{k-2} 与 A 相似且不改变对角线或第一非对角线元素. 最后可用对角线相似变换 D 使 A_{k-2} 的第一非对角线元素等于 1，且不改变对角线元素. 这样，$Q = \tilde{Q}D$ 即为将 A 变换成 Jordan 型所需要的变换. 这就是所要证明的.

如果对特殊的 p 值选取 $\mathbf{l}$ 达到给定的特征值组，则对其他 p 值给出同样特征值的 $\mathbf{l}$ 的分量容易按如下的方法寻找.

$k \times k$ Pascal 矩阵 A 的 $(k-p)\times(k-p)$ 下主子式在对角线变换

$$D_{pk} = \operatorname{diag}\left[d_i = \binom{p+i}{i}\right]$$

下与 $A_{0,k-p}$ 相似.如果对某个 p 来说，$\mathbf{l}$ 的最后 $k-p$ 个元称为 $\mathbf{l}_{pk}$，通过线性变换，可以从 $\mathbf{l}_{0k-p}$ 找到 $\mathbf{l}_{pk}$. 记 $S_{pk} = (I - \mathbf{l}_{pk}\boldsymbol{\delta}_0^T)A_{pk}$，有

$$\begin{aligned} D_{pk}S_{pk}D_{pk}^{-1} &= (I - D_{pk}\mathbf{l}_{pk}\boldsymbol{\delta}_0^T D_{pk}^{-1})D_{pk}A_{pk}D_{pk}^{-1} \\ &= (I - D_{pk}\mathbf{l}_{pk}\boldsymbol{\delta}_0^T)A_{0,k-p}, \end{aligned}$$

所以 $\mathbf{l}_{pk} = D_{pk}^{-1}\mathbf{l}_{0,k-p}$.

i 的 P 个元素仍要选取，对于任何选取方法仍将是稳的，因为它们不影响特征值. 在下一节我们将看到可选它们来改进方法的阶，这一节的结果的重要性在于它说明对于稳定多

步法的限制，在多值方法中是可以克服的.

10.4. 标准形式多值方法阶的改进

我们考虑 A 是 $k \times k$ Pascal 三角形矩阵的多值方法. 设 $\mathbf{l}$ 的最后 $k-p$ 个元素已经选定，给出了所需要的稳定性性质. 可以选取 $\mathbf{l}$ 的其余 p 个元素，以便得到可能的最高阶. 为做到这一点，首先重新定义截断误差. 在用方程(10.18)和(10.19)的阶的定义中，假定对标准形式方法 $\mathbf{a}$ 含有元素 $h^q y^{(q)}/q!$. 于是，误差不小于第一个略去的项 $O(h^k)$. 下面将看到，如果采用另外的定义，更高的阶是可能的.

定义 $(k'-k)$ 维向量 $\mathbf{r}(t)$ 为

$$\left[\frac{h^k y^{(k)}}{k!}, \cdots, \frac{h^{k'-1} y^{(k'-1)}}{(k'-1)!}\right]$$

的转置，并令 $\mathbf{a}^C(t)$ 为向量 $[y, hy', \cdots, h^{k-1}y^{(k-1)}/(k-1)!]$ 的转置，并在 t 处的微分方程的解上求值. 假定我们要计算的向量 $\mathbf{a}(t)$ 由

$$\mathbf{a}(t) = \mathbf{a}^C(t) + E\mathbf{r}(t) \tag{10.56}$$

给出，其中 E 是 $k \times (k'-k)$ 常矩阵. 如果从方法 $\mathbf{a}(t_n - h)$ 计算 $\tilde{\mathbf{a}}(t_n)$，对解在 $C_{k'+1}$ 中的方程，它与正确值有如下关系：

$$\tilde{\mathbf{a}}(t_n) = \mathbf{a}(t_n) + \mathbf{d}_n h^{k'} + O(h^{k'+1}). \tag{10.57}$$

于是，截断误差定义成 $\mathbf{d}_n h^{k'} + O(h^{k'+1})$，并称方法的阶为 $k'-1$.

这表示我们计算的向量 $\mathbf{a}$ 含 k 阶和更高阶的导数分量，与前面取 $O(h^k)$ 的值是不同的. 事实上，这个定义表示起始误差和步长改变误差的阶数比截断误差低. 使用者未必计算含这些分量的起始值，而以 $O(h^k)$ 的误差开始积分. 还有改变步长的简单方法，即左乘对角线矩阵，不保留 E，因此，也引进同样阶的误差.

如果在定理 10.8 中采用这个新的 $\mathbf{a}(t)$ 的定义，它的证明不变．倘若起始误差恰好是这样的阶，方法整体的收敛速度将为 $O(h^{k'-p})$.

令 $(k'+1)\times(k'+1)$ Pascal 矩阵按

$$
\begin{array}{c}
k \\ \hline k'-k \\ \hline 1
\end{array}
\left[\begin{array}{c:c:c}
A & D & \mathbf{c}_1 \\ \hdashline
0 & B & \mathbf{c}_2 \\ \hdashline
0 & 0 & 1
\end{array}\right]
$$

分块．如果解在 $C_{k'+1}$ 中，则由 Taylor 级数得

$$
\begin{aligned}
\mathbf{a}^C(t) = A\mathbf{a}^C(t-h) + D\mathbf{r}(t-h) \\
+ \mathbf{c}_1 a_{k'} + O(h^{k'+1})
\end{aligned}
\qquad (10.58)
$$

和

$$
\mathbf{r}(t) = B\mathbf{r}(t-h) + \mathbf{c}_2 a_{k'} + O(h^{k'+1}),
$$

其中 $a_{k'} = h^{k'}y^{(k')}(t-h)/k'!$．为了简化推导，我们只考虑一次校正迭代，所以

$$
\tilde{\mathbf{a}}(t) = A\mathbf{a}(t-h) + \mathbf{l}F(A\mathbf{a}(t-h)). \qquad (10.59)
$$

将 (10.56) 和 (10.57) 代入 (10.59)，得到

$$
\begin{aligned}
&\mathbf{a}^C(t_n) + E\mathbf{r}(t_n) + \mathbf{d}_n h^{k'} + O(h^{k'+1}) \\
&\quad = A(\mathbf{a}^C(t_n-h) + E\mathbf{r}(t_n-h)) \\
&\qquad + \mathbf{l}F(A\mathbf{a}^C(t_n-h) + AE\mathbf{r}(t_n-h)).
\end{aligned}
\qquad (10.60)
$$

将 (10.58) 代入 (10.60) 并利用 $F(\mathbf{a}^C) = 0$ 得到

$$
\begin{aligned}
\mathbf{d}_n h^{k'} = &\left[\left(I + \mathbf{l}\frac{\partial F}{\partial \mathbf{a}}\right)(AE - D) - EB\right]\mathbf{r}(t_n - h) \\
&- \left[E\mathbf{c}_2 + \left(I + \mathbf{l}\frac{\partial F}{\partial \mathbf{a}}\right)\mathbf{c}_1\right]a_{k'} + O(h^{k'+1}).
\end{aligned}
\qquad (10.61)
$$

但是

$$
\frac{\partial F}{\partial \mathbf{a}} = -\boldsymbol{\delta}_p^T + \sum_{q=1}^{p}\frac{(p-q)!\,h^q f_{y(p-q)}}{p!}\boldsymbol{\delta}_{p-q}^T.
$$

为了满足 (10.61)，含 h^q 的项 $(q < k')$ 必须为零，而 $\mathbf{d}_n$

由 $h^{k'}$ 的项给出．因此

$$R=(I-\mathbf{I}\boldsymbol{\delta}_p^T)(AE-D)-EB=0, \quad (10.62)$$

$$d_n=-\frac{[E\mathbf{c}_2+(I-\mathbf{I}\boldsymbol{\delta}_p^T)\mathbf{c}_1]y^{(k')}}{(k')!}$$

$$+\mathbf{I}\sum_{q=1}^{\tilde{p}}\frac{(p-q)!f_{y^{(p-q)}}}{p!(k'-q)!}\boldsymbol{\delta}_{p-q}^T(AE-D)\boldsymbol{\delta}_{k'-k-q}y^{(k'-q)}, \quad (10.63)$$

其中 $\tilde{p}=\min(p,k'-k)$,

$$\sum_{q=1}^{k'-k-i-1}\frac{(p-q)!f_{y^{(p-q)}}}{(k'-q)!}\boldsymbol{\delta}_{p-q}^T(AE-D)\boldsymbol{\delta}_{k'-k-q-i-1}y^{(k'-q-i-1)}=0,$$
$$i=0,1,\cdots,k'-k-2. \quad (10.64)$$

如果记

$$\mathbf{u}^T=\boldsymbol{\delta}_p^T(AE-D), \quad (10.65)$$

方程 (10.62) 变成

$$R=AE-EB-\mathbf{I}\mathbf{u}^T-D=0. \quad (10.66)$$

注意 A 和 B 是对角线上元素为 1 的上三角形矩阵，我们看到 R 的第一列给出

$$\sum_{j=0}^{k-1}(A_{ij}-\delta_{ij})E_{j0}-l_iu_0=D_{i0}, \quad 0\leqslant i\leqslant k-1. \quad (10.67)$$

一个强稳定方法在单位圆上无附加根，所以 $l_{k-1}\neq 0$．因此，当 $i=k-1$ 时，(10.67)给出对 u_0 的解；当 $p\leqslant i\leqslant k-2$ 时，给出对 $E_{i+1,0}$ 的解．类似地，R 的第 m 列后 $k-p$ 个元素可按 $m=1,2,\cdots,k-k'-1$ 对 u_m 和 $E_{i+1,m}$, $p\leqslant i\leqslant k-2$, 解出．现在 $\mathbf{u}^T$ 已知，(10.65) 提供了对 $E_{p,i}(0\leqslant i\leqslant k'-k-1)$的非奇异方程．

由于 u_0 不为零，(10.66) 的前 p 行构成方程组，它可以从最左边开始到对角线逐次求解，即

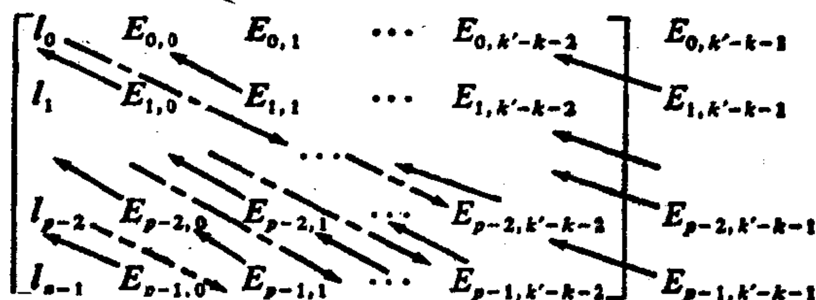
$$R_{p-1,0}, R_{p-1,1}, R_{p-2,0}, R_{p-1,2}, \cdots.$$

在这种过程中,可以无约束地选取 E 的最后一列前 p 个元素.特别,如果 $k'-k \leqslant p$,它们可以如此选择,使 E 顶上的行为零,所以,至少函数值不含有附加的导数值. 这个过程还可以对 $i = p-k'+k, \cdots, p-1$ 确定 l_i 的值;其余的分量 l_i 由

$$E_{k'-k-1,i+k'-k}, \quad i = 0, \cdots, p-k'+k-1$$

的值确定.

E 和 $\mathbf{I}$ 的元素对其它元素的依赖性可从下面的图中看出.由 $R_{ij} = 0$ 可以求得第 (i, j) 位置上的元素为 1,这些元素按箭头所示的次序处理,每次从右边的列开始一个新的对角线,并且引进一个自由参数 $E_{i,k'-k-1}$:



一般,(10.64) 只当 $AE-D$ 的适当的下三角形部分为零或者 f 的适当偏导数处处为零时才能满足. 前面的条件与那些仍为自由的参数(E 的最后一列)是无关的,所以它未必能满足.

于是,我们证明了下面的定理.

定理 10.12. 如果使 $\partial f/\partial y^{(p-i)} \neq 0$ 的 i 的最小值为 q,则存在阶为 $k+q-1$ 的强稳定 k 值方法.

特别,对特殊的 p 阶方程 $y^{(p)} = f(y, t)$,存在阶为 $k+p-1$ 的方法.注意,对 $p=1$,将 $k-1$ 步的 Adams-Bashforth-Moulton 方法表成 k 值方法有阶为 $k-1$ 的预估 [超过 $y^{(k-1)}$ 的项不计在内],但在 E 为 $k \times 1$ 矩阵的意义下,它是 k 阶方

法.因此,用这种方法计算的 $\mathbf{a}$ 除了在零位置上外,含有 $h^k y^{(k)}$ 的固定倍数.如果 $\mathbf{a}_0$ 的初始值为 $h^q y^{(q)}/q!(0 \leqslant q < k)$, 则存在一个量级为 $O(h^k)$ 的起始误差(定理 10.7). 类似地, 步长改变固定次数只引起 $O(h^k)$ 的误差, 但是如果步长改变数 $= O(h^{-1})$, 将得到 $O(h^{k-1})$ 级的全体误差.

如果用 M 次校正迭代, 方程 (10.61) 要修改,即将 $(I - \mathbf{l}(\partial F/\partial \mathbf{a}))$ 都换成

$$\prod_{i=1}^{M} \left[I + \mathbf{l}\left(\frac{\partial F}{\partial \mathbf{a}}\right)_i \right],$$

其中足标 i 表示偏导数对不同的迭代 i 是在不同的点上计算的. 如果 $l_p \neq 0$, $\mathbf{l}$ 换成 $\tilde{\mathbf{l}} = \mathbf{l}(1-(1-l_p)^M)/l_p$;如果 $l_p = 0$, $\mathbf{l}$ 换成 $\mathbf{l}M$, 这一节的结论仍不变.

10.5. 误差的渐近性质

用单步方法,我们能将误差表示成

$$e_n = h^r \delta(t_n) + O(h^{r+1}).$$

在这一节,我们研究对多步方法与多值方法的类似的结果.先考虑一阶方程只用校正的多步方法. 讨论的方法与定理 1.3 中对 Euler 方法所使用的技巧是等同的. 我们有

$$\sum_{i=0}^{k} (\alpha_i y_{n-i} + h\beta_i f_{n-i}) = 0 \qquad (10.68)$$

和

$$\sum_{i=0}^{k} (\alpha_i y(t-ih) + h\beta_i f(y(t-ih)))$$

$$= C_{r+1} h^{r+1} y^{(r+1)}(t) + O(h^{r+2}), \qquad (10.69)$$

相减得到

$$\sum_{i=0}^{k}\left[\alpha_i e_{n-i} + h\beta_i f_y(\xi_{n-i})e_{n-i} + \frac{C_{r+1}}{\sum_{j=0}^{k}\beta_j}\beta_i h^{r+1}y^{(r+1)}(t_n)\right]$$

$$= O(h^{r+2}).$$

令 $e_n = h^r\delta_n$ 并注意

$$f_y(\xi_{n+i}) = f_y(y(t_{n-i})) + O(h^r),$$
$$\beta_i y^{(p+1)}(t_n) = \beta_i y^{(p+1)}(t_{n-i}) + O(h),$$

得到

$$\sum_{i=0}^{k}\left[\alpha_i\delta_{n-i} + h\beta_i(f_y(y(t_{n-i}))\delta_{n-i} + \frac{C_{r+1}}{\Sigma\beta_j}y^{(r+1)}(t_{n-i}) + O(h))\right] = 0. \quad (10.70)$$

这是用我们正在分析的多步方法得到的一个方程的解，这方程与

$$\delta'(t) = g(t)\delta(t) - \frac{C_{r+1}}{\Sigma\beta_j}y^{(r+1)}(t) \quad (10.71)$$

的差为 $O(h)$，其中 $g(t) = f(y(t))$. 如果起始值是精确的，由定理 10.6 和（10.71）为适定的事实，(10.70) 的解收敛于 (10.71) 的解，我们得到

定理 10.13. 对一阶方程的收敛多步方法

$$e_n = h^r\delta(t_n) + O(h^{r+1}),$$

其中起始值若是精确的，$\delta(t)$ 满足 (10.71) 且 $\delta(0) = 0$. 我们也看到，使 $\Sigma\beta_j = 1$ 归一化的理由.

一般，起始值是不精确的. 在定理 10.7 中我们看到，如果起始误差是 $O(h^{r'})$，存在一个量级为 $O(h^{r'})$ 的附加误差. 欲知这些对误差的渐近影响，特别是在 $r' \leqslant r$ 的情形，必须假定实际起始误差为 $\bar{\delta}_i h^{r'} + O(h^{r'+1})$ $(0 \leqslant i < k)$，其中 $\bar{\delta}_i$ 是常数. 因此，考察 (10.70) 的具有初始条件 $\delta_i = \bar{\delta}_i h^{r'-r}$ 的解. 如果 $r' > r$，渐近误差如定理 10.13 所给定；如果 $r' <$

r，则误差形如

$$e_n = \delta_n h^{r'} + O(h^{r'+1}),$$

其中 δ_n 由 $C_{r+1}=0$ 和起始值 $\delta_i=\tilde{\delta}_i$ 的 (10.70) 给出;如果 $r'=r$，则从起始值 $\delta_i=\tilde{\delta}_i$ 解 (10.70).

定理 10.14. 令 $\rho(\xi)=0$ 的根为 $\xi_1=1, \xi_2, \cdots, \xi_k$，数 $z_{ij}(0\leqslant j<k, 1\leqslant i\leqslant k)$ 由

$$z_{ij}=\begin{cases} j^m \xi_i^j, & \text{如果 } \xi_i \neq 0, \\ \delta_{ij}+1, & \text{如果 } \xi_i = 0 \end{cases}$$

定义,其中,当根 ξ_i 第一次出现时 m 为零,第二次出现时为 1，等等. 如果 u_i 由

$$\tilde{\delta}_j=\sum_{i=1}^{k} u_i z_{ij}, \qquad 0\leqslant j<k$$

确定[若 $\{z_{ij}\}$ 非奇异,这永远可能],于是若 $r'=r$,则强稳定收敛方法的误差 e_n 满足

$$e_n = h^{r'}\delta(t_n) + O(h^{r'+1}),$$

其中 $\delta(t)$ 为 (10.71) 中 $\delta(0)=u_1$ 的解. 如果 $r'<r$, $\delta(t)$ 由同一个问题但其中 $C_{r+1}=0$ 给出.

证明可在 Henrici (1962) 的 5.3 节中找到,那里给出了比单位圆上存在附加根的性质更强的结果. 其影响是引进另外的分量 $h^{r'}u_i\delta_i(t_n)$，其中 $\delta_i(t_n)$ 是相应微分方程的解. 由于不推荐弱稳定方法,细节就不讲了.

现在研究 p 阶方程多值方法的误差的渐近形式. 这里仅考虑用一次校正迭代的方法,否则推导会变得不必要的复杂. 这里给出的结果是定理 10.13 的自然推广，即

定理 10.15. 如果收敛的多值方法的截断误差如下:

$$\left[\mathbf{d}_2 y^{(k')} + \mathbf{I}\sum_{q=1}^{p} \frac{\partial f}{\partial y^{(p-q)}} y^{(k'-q)} d_{1q}\right] h^{k'} + O(h^{k'+1}) \tag{10.72}$$

[参看方程 (10.63)], 而 $\{\nu_m\}$ 由

$$\mathbf{d}_2 = \sum_{m=0}^{k-1} \gamma_m S^m \mathbf{l} \tag{10.73}$$

确定,其中

$$S = (I - \mathbf{l}\boldsymbol{\delta}_p^T)A,$$

则从精确初始条件出发, 积分中的误差在 y_n 上有误差分量 $h^{k'-p}\delta(t_n) + O(h^{k'-p+1})$, 其中 $\delta(t)$ 是

$$\delta^{(p)}(t) = \sum_{q=1}^{p} \frac{\partial f}{\partial y^{(p-q)}}(\delta^{(p-q)} + d_{1q}y^{(k'-q)}(t)) + \sum_{m=0}^{k-1} \gamma_m y^{(k')}(t) \tag{10.74}$$

具有 $\delta(0) = \delta^{(1)}(0) = \cdots = \delta^{(p-1)}(0) = 0$ 的解. 如果方法是强稳定的,则可看出 γ_m 满足 (10.73).

证明. 观察 $\boldsymbol{\delta}_n = h^{-k'+p}\mathbf{e}_n$, 其中 $\mathbf{e}_n = \mathbf{a}_n - \mathbf{a}^C(t_n) - E\mathbf{r}(t_n)$ [见方程 (10.56)],

$$\boldsymbol{\delta}_{n,(0)} = A\boldsymbol{\delta}_{n-1}, \tag{10.75}$$

$$\boldsymbol{\delta}_n^{*} = \boldsymbol{\delta}_{n,(0)} + \mathbf{l}\,\frac{\partial F}{\partial \mathbf{a}}\boldsymbol{\delta}_{n,(0)} + \mathbf{l}h^p \sum_{q=1}^{p} \frac{\partial f}{\partial y^{(p-q)}} y^{(k'-q)}(t_n)d_{1q} + h^p \sum_{m=0}^{k-1} \gamma_m S^m \mathbf{l} y^{(k')}(t_n) + O(h^{p+1}). \tag{10.76}$$

不把 $h^p\gamma_m S^m \mathbf{l} y^{(k')}(t_n)$ 加到 $\boldsymbol{\delta}_n$, 而将 $h^p\gamma_m \mathbf{l} y^{(k')}(t_n - mh) + O(h^{p+1})$ 加到 $\boldsymbol{\delta}_{n-m}$, 具有同样的作用. 如果这样做了,下面的步骤还要将 γ_i "往后退". 所以,如果 S^m 已经从 (10.76) 中去掉, 则在 $O(h)$ 的范围内, (10.75) 和 (10.76) 的解是相同的 (由于退到负 n 在初值上的变化和由较大的 n 退回在 t_n 上值的变化都是 p 阶的, 都比解小, 所以它们可以不考虑). 作这样的改变后, (10.75) 和 (10.76) 都是差分方程. 如果 (10.74) 用所讨论的方法求解,这差分方程就得到了. 于是,结果的第

一部分成立.

第二部分由 $S^m\mathbf{l}(m=0,1,\cdots,k-1)$ 为线性无关的事实得到. 我们可以这样看出，由于对强稳定方法 $l_{k-1}\neq 0$，$\mathbf{l}$ 是对应于 A 的特征值 1 的秩为 k 的主向量 [即 $(A-I)^m\mathbf{l}=0\Longrightarrow m\geqslant k$.]，展开后，得到

$$S^m\mathbf{l}=A^m\mathbf{l}+w_{m1}A^{m-1}\mathbf{l}+\cdots+w_{mm}\mathbf{l}.$$

每个新的 $A^m\mathbf{l}$ 引入一个 $S^m\mathbf{l}(m=0,\cdots k-1)$ 的新的主向量分量，因此，$S^m\mathbf{l}$ 是线性无关的.

问　　题

1. 假定用 (10.2) 和 (10.3) 给出的并且 $\alpha_0=-1,\beta_0=\beta_{q0}=0$ 的预估公式，然后用 (10.4) 和

$$\frac{h^q y^{(q)}_{n,m+1}}{q!}=\sum_{i=1}^{k}\left(\alpha^*_{qi}y_{n-i}+\beta^*_{qi}\frac{h^p}{p!}f_{n-i}\right)$$
$$+\beta^*_{q0}\frac{h^p}{p!}f(y_{n,(m)},\cdots,y^{(p-1)}_{n,(m)},t_n)$$

校正，给出用 $\alpha^*_{qi},\beta^*_{qi},\alpha^*_i,\beta^*_i,\alpha_i$ 和 β 表示的系数 α_{qi} 和 β_{qi} 的表达式，它使得用向量

$$\mathbf{y}_n=\left[y_n,y_{n-1},\cdots,y_{n-k+1},hy'_n,\frac{h^2y''_n}{2},\cdots,\frac{h^py^{(p)}_n}{p!},\cdots,\frac{h^py^{(p)}_{n-k+1}}{p!}\right]$$

将方法表示成矩阵形式成为可能.

2. 证明引理 10.1 中确定的 a_{ij} 的矩阵是非奇异的. [提示: 寻找出现在它的行列式中形为 $(\xi^p_i-\xi^q_i)$ 的因子]

3. 如果对方程

$$y'''=f(y,y',t)$$

应用 r 阶预估公式和 s 阶校正公式，那么对这个方程 M 次校正迭代后方法的阶是多少?

4. 证明定理 10.3.

5. 考虑

$$y^{(p)} = \frac{k!\,t^{k-p}}{(k-p)!}.$$

证明引理 10.4。

6. 通过考虑 $y^{(p)} = y^{(p-1)}$，证明预估的阶加上 M（校正迭代的次数）不能小于对 p 阶方程的方法的阶 r。

7. 对预估和校正均是二阶的，且仅作一次校正迭代的 Adams–Bashforth–Moulton 格式寻找定理 10.5 证明中定义的 $\tilde{S}_n$。问：增加一次迭代会有什么差别？

8. 考虑下面的方法

$$y_{n+\frac{1}{2}} = y_n + \frac{5}{24}hf(y_n) + \frac{h}{3}f(y_{n+\frac{1}{2}}) - \frac{h}{24}f(y_{n+1}),$$

$$y_{n+1} = y_n + \frac{h}{6}f(y_n) + \frac{2}{3}hf(y_{n+\frac{1}{2}}) + \frac{h}{6}f(y_{n+1}).$$

(a) 为确定 $y_{n+\frac{1}{2}}$ 和 y_n，用 Euler 方法估计 $f(y_{n+\frac{1}{2}})$ 和 $f(y_{n+1})$。

(b) 用某种方法（这是无关紧要的）精确求解方程。

问在每一种情形中方法的阶是多少？

9. p 阶方程的强稳定方法满足严格根条件，考虑特殊的 p 阶方程和定理 10.12 的结果。在定理 10.12 证明中一个注表明，可以选择 E 使 $E_{0,i} = 0\ (0 \leq i \leq k' - k - 1)$。试说明如何改变定理 10.8 和的证明10.9 才能证明如下事实：假定a_0为 $h^q y^{(q)}/q!\,(0 \leq q < k')$ 的精确值，则全体误差为 $O(h^k)$。

10. （非常复杂！）

(a) 对M次迭代方法，推导定理 10.12前面的方程。如果一次迭代方法达到的阶为 $k + q - 1$，问在以后的迭代中，这个阶能保持吗？截断误差是多少？

(b) 证明定理 10.15 对M次校正迭代的推广。

11. 证明稳定显式 k 步方法的最高阶为 k（提示：应用当 $\mu \geq 0$ 时 $\sum_{\nu=0}^{\mu} c_\nu > 0$ 的事实。也请证明这一点）。

11. 特殊问题的特殊方法

这一章的大部分内容是处理Stiff微分方程的．在许多用计算机作为辅助设计技术，特别是网络分析和进行模拟时，会出现这些方程．这些方程也出现在几乎所有化学动力学的研究中．我们还要讨论与其密切相关的一个问题，即与微分方程同时解一组非线性代数方程的问题．这些问题常常在网络分析和模拟时出现．最后，我们还要简短地讨论当微分方程的解在一些点上已知时，寻找未知参数的值的问题，这是一个积分化学动力学方程常常会遇到的问题．

11.1. Stiff 方程

在实际方程中，由于存在相差很大的时间常数，Stiff微分方程是经常出现的．所谓时间常数是工程师和物理学家用来表示衰减速度的术语．例如，方程 $y'=\lambda y$ 有解 $Ce^{\lambda t}$．如果 λ 是负的，则在时间 $-1/\lambda$ 内，y 衰减 e^{-1} 倍．$-\dfrac{1}{\lambda}$ 就是时间常数．λ 负得愈多，时间常数愈小．实际的系统经常是以指数函数的方式变化的，至少在局部范围是这样（电容器的放电，化学反应趋于稳定等）．在一个系统中，不同的项以不同的速度衰减．对于系统

$$\mathbf{y}'=f(\mathbf{y}),$$

衰减速度局部地可与 $\partial f/\partial y$ 的特征值有关．如果一些反应是慢的，而另外一些反应是快的，则快的部分将决定方法的稳定性，虽然这时这些分量可以衰减到微不足道的程度，使得方法

的截断误差由变化慢的分量来确定．例如，考虑“系统”

$$y' = -y, \qquad y(0) = 1,$$
$$z' = -1000z, \quad z(0) = 1. \tag{11.1}$$

事实上，这两个方程是相互独立的，我们可以分别来考察它们每一个的性质．如果用研究过的大多数方法来考察，将会发现，稳定性要求量 $1000h$ 小于某一个界．Euler 法要求它小于 2，以便满足 $|1 + \lambda h| < 1$，而四阶 Runge-Kutta 方法大约要求它小于 2.8．由图 8.2 看到，Adams-Moulton 方法的高阶方法对 λh 甚至有更多的限制．如果方程 (11.1) 用这些方法来积分，只能采用比 z 的时间常数大得不太多的数作为步长．积分几步之后，z 的值将小到与 y 相比可以忽略的程度．虽然只在第一个分量中含有有效的信息，但是由这点往后，由于第二个分量的缘故，仍必须采用非常小的步长．这就是 Stiff 方程问题．

在上面的特殊例子中，两个分量是可以分离的，并且对其中的每一个可以应用不同的方法或步长．按照这一点已提出许多建议，它们在速变分量变化区域中有效．但是，将一般方程分离成二个或更多个简单分量是不可能的．

例如，考虑

$$u' = 998u + 1998v, \qquad u(0) = 1,$$
$$v' = -999u - 1999v, \quad v(0) = 0 \tag{11.2}$$

它们是由 (11.1) 经变换

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix}$$

导出的，因此，两个分量有类似的性质．它们的解为

$$u = 2e^{-t} - e^{-1000t},$$
$$v = -e^{-t} + e^{-1000t},$$

所以，两个因变量都含有快变和慢变的分量．它们的解在图

11.1 中表出．方程(11.2)即使在快变分量很小时，也产生对 h 的同样的限制．如果可以找到由 (11.2) 到 (11.1) 的逆变换，则各个分量可以分别处理．但是对大的方程组，或者变换依赖于 t 时，这不是简单的工作．

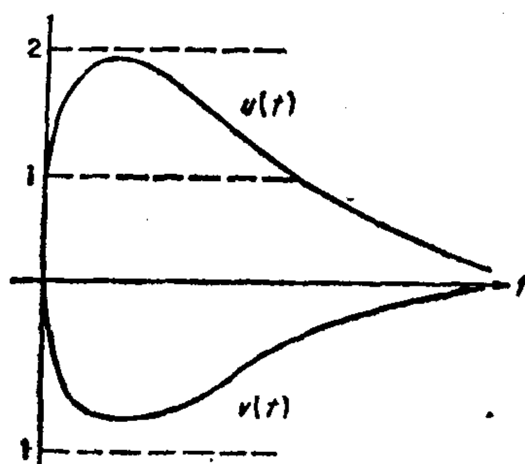


图 11.1. Stiff 问题

为了观察这种问题，不一定考虑方程组，简单的例子为

$$y' = \lambda(y - F(t)) + F'(t). \tag{11.3}$$

如果 $F(t)$ 是光滑的慢变函数，而 $\lambda \ll 0$，则它就具有类似的性质．(11.3) 的解为

$$y = (y_0 - F(0))e^{\lambda t} + F(t), \tag{11.4}$$

甚至当 $y_0 - F(0) \neq 0$ 时，λt 很快取充分大的负数，使得第一项与第二项相比足够小．如果用讨论过的任何单步或多步方法考察(11.3)的误差方程，我们将会看到，局部截断误差由 h 和 F (使 $\lambda t \ll 0$ 的时间导数) 来确定，而稳定性依赖于 $h\lambda$ 的值．

在象

$$\mathbf{y}' = A(\mathbf{y} - F(t)) + F'(t) \tag{11.5}$$

一样的方程组中，A 的特征值起 λ 的作用．在这种情形，我们必须考虑复数 λ．如果 A 的所有特征值均有负实部，则当 t 趋

于无穷大时，(11.5) 的解收敛于 $F(t)$.

虽然利用我们讨论过的任何一种方法得到的 (11.3) 的数值近似解当 $h\to 0$ 时确实收敛到它的解，但是，为实际达到要求的精度，h 必须非常非常小，小到舍入误差和计算时间问题变得相当严重. 因此，现在的问题是要推导保证稳定性但不限制步长的方法.

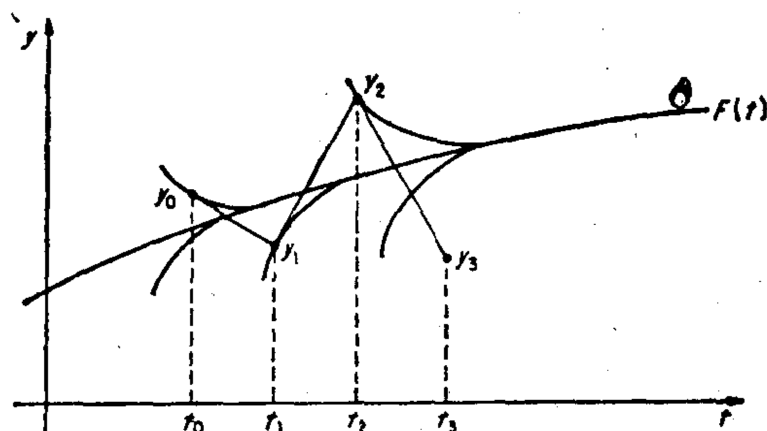


图 11.2. Stiff 问题的 Euler 方法

当解中类似 $Ce^{\lambda t}$ 的一些项比起其它项十分小，而且它们仍在继续衰减，对 $e^{\lambda h}$ 的逼近就不必很精确，只要逼近的误差小于 1，使得这些项不扩大即可. 图 11.2 说明对问题 (11.3) 使用 Euler 法的结果. 如果 $|1+h\lambda|>1$，误差每一步均是扩大的.

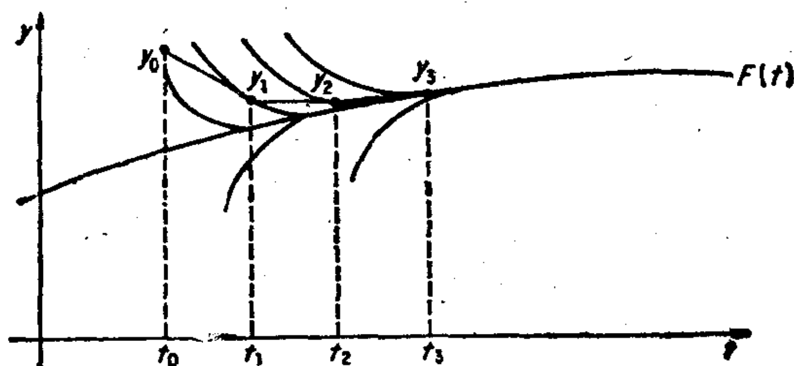


图 11.3. Stiff 问题的向后 Euler 方法

图 11.3 表明，对同一个问题使用向后 Euler 方法 $y_{n+1}=y_n+hf(y_{n+1}, t_{n+1})$ 的结果. 这是一个隐式方法，对 y 的线性方程组可直接解出. 从图中可以看到，虽然表示 $e^{\lambda t}$ 项的精度差，但它是稳定的. 通过分析可以知道，误差每一步均扩大 $(1-h\lambda)^{-1}$ 倍. 如果 $\mathrm{Re}(\lambda)<0$, 则这个数小于1. 显然，这是一个很好的性质，Dahlquist (1963) 将其总结成为 A 稳定性.

定义 11.1. 方法称为 A 稳定的，如果将它以正固定步长 h 应用到具有负实部的（复）常数 λ 的微分方程 $y'=\lambda y$ 上，得到的数值近似当 $n\to\infty$ 时都趋于零.

这一节只讨论一阶方程，虽然对高阶方程显然也有类似的性质. 其中有些性质在 11.2 节中考察. 在 Bjurel 等 (1970) 中给出了产生 Stiff 问题的各个应用领域以及求其解的方法的详尽评论.

11.1.1. 多步方法

在关于 A 稳定性的最早一批重要工作中，Dahlquist(1963) 还提供了一个结果，即证明 A 稳定多步方法的阶不能超过 2，以及阶为 2 且具有最小误差常数 c_3 的方法是梯形法，其中 $c_3=\frac{1}{12}$. 这是一个具有约束性的结果，它表明如果要用多步方法，A 稳定性的要求必须放宽. 将多步方法应用到线性问题 $\mathbf{y}'=A\mathbf{y}$ 上，其稳定性由

$$\rho(\xi)+h\lambda_i\sigma(\xi)=0,\ i=1, 2, \cdots, s$$

的根来确定，其中 λ_i 是 A 的特征值. 由于这些值是固定的，我们不需要整个负半平面上的稳定性，而只要在 $h\lambda_i$ 所占有的区域中稳定即可. 由图 11.4 所示，对于慢变化的 A，这些区域组成 $h\lambda$ 平面上的一列楔形. Widlund (1967) 定义 $A(\alpha)$ 稳定性如下：

定义 11.2. 一个方法称为 $A(\alpha)$ 稳定的，$\alpha\in(0,\pi/2)$ 如果以固定的h，对所有满足 $|\arg(-\lambda)|<\alpha$，$|\lambda|\neq 0$ 的方程 $y'=\lambda y$ 的数值近似，当 $n\to\infty$ 时收敛于零.

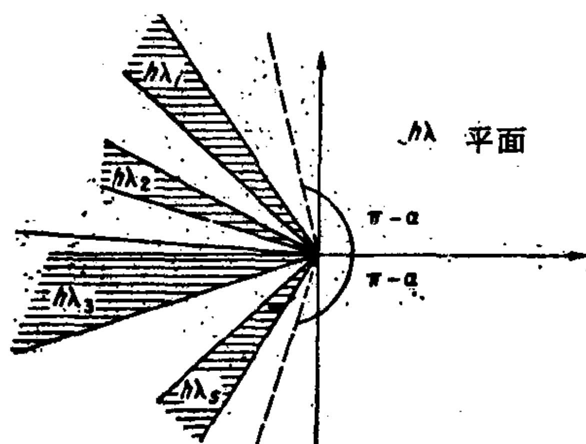


图 11.4. $A(\alpha)$ 稳定性区域

这表示绝对稳定性区域包含图 11.4 中虚线左面的楔形. Widlund 接着证明了对任意 $\alpha<\frac{\pi}{2}$ 和 $r\leqslant 4$，存在 $A(\alpha)$ 稳定的 r 步 r 阶方法. 这样，对于给定的其解为渐近稳定的线性问题（即 A 的特征值均严格地在负半平面中），存在对任意（正）h 为稳定的阶为 4 或小于 4 的多步方法. 由于 $A\left(\frac{\pi}{2}\right)$ 稳定方法是 A 稳定的，阶 $r\geqslant 3$ 的 $A(\alpha)$ 稳定方法的截断误差当$\alpha\to\frac{\pi}{2}$ 时无限增大.

另一个将 A 稳定性减弱的概念定义成 Stiff 稳定性 [Gear (1969)]. 图 11.5 定义 $h\lambda$平面上的区域.

定义 11.3. 一个方法是 Stiff 稳定的，如果在区域 $R_1(\mathrm{Re}(h\lambda)\leqslant D)$中，它是绝对稳定的，而在区域 $R_2(D<\mathrm{Re}(h\lambda)<\alpha, |\mathrm{Im}(h\lambda)|<\theta)$ 中，它是精确的.

提出这个定义的理由如下：$e^{h\lambda}$ 是一步中对特征值 λ 的分量的改变量，如果 $h\lambda=u+iv$，则改变量的量级为 e^u. 如果 $u<D<0$，则在一步中这分量至少减小到 e^D 倍. 我们不考虑那些非常小的分量的精确度，所以，对某个 D，宁愿忽略 R_1 中的所有分量. 这时，我们恰好需要方法是绝对稳定的. 在原点附近，要考虑精度，这时需要相对的或绝对的稳定性.

如果 $u > \alpha > 0$，在每一步这分量至少增加 e^{α} 倍．我们必须对此加以限制，因为要适应这个改变量，节点要充分精细，因此，我们不使用区域 $u > \alpha$．如果 $|v| > \theta$，在每一步至少有 $\theta/2\pi$ 个完整的振动．除了不考虑衰减分量的区域 R_1 和不用的区域 $u > \alpha$ 外，我们必须描述这些分量的变化．大家都知道，为了表示频带限制的信号，至少对出现的最高频率每一周取两个样点．事实上，为了数值精度，大约上述数字的五倍是必要的，所以，θ 一定小于 $\pi/5$．

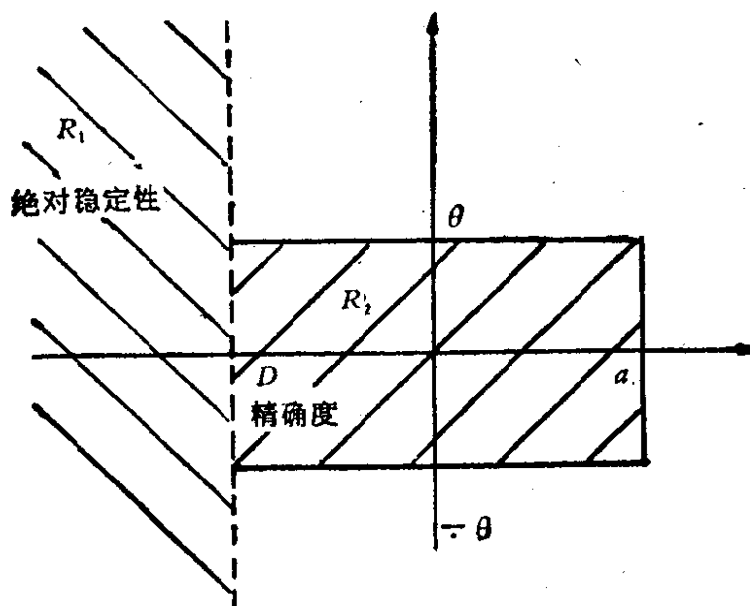


图 11.5． Stiff 稳定性

自然我们要问，是否存在阶大于 2 的 Stiff 稳定方法．在 Gear (1969) 中，具有 $\sigma(\xi) = \xi^k$ 的 k 阶 k 步方法被证明对某些 D, α 和 θ 当 $k \leqslant 6$ 时是 Stiff 稳定的．先由 $\sigma(\xi)$ 计算 $\rho(\xi)$，以便得到阶为 k 的方法，然后可以得到这个方法（见 8.1.1 节）．于是，在 μ 平面上使 $\rho(\xi) + \mu\sigma(\xi) = 0$ 的根的量值为 1 的轨迹，可由描绘 $\mu = -\rho(e^{i\theta})/\sigma(e^{i\theta}), \theta \in [0, 2\pi]$，来画出．这些轨迹在图 11.6 中由 $k=1, 2, 3$ 和图 11.7 中由 $k=4, 5, 6$ 表出．在 $\mu = +\infty$，$\rho(\xi) + \mu\sigma(\xi) = 0$ 的根为 $\sigma(\xi) = 0$ 的

根，或者都是零．闭合轨迹外的所有点由连续曲线与 $\mu=+\infty$ 连接．由于根是 μ 的连续函数，对于轨迹外对 μ 的所有根，其数值均小于 1．因此，绝对稳定区域是闭合曲线的外部．对于 $k=7, 8, \cdots, 15$，这些方法不是 Stiff 稳定的．这些方法在 Henrici（1962）5.1.4 节中给出，在那里称为"基于数值微分的方法"．前两种方法（$k=1,2$）对 Stiff 方程的可用性由 Curtiss 和 Hirschfelder（1952）指出．对于 $k=1$，我们得到向后 Euler 方法．最近，Stiff 稳定的七阶和八阶多步方法［Dill（1969）］

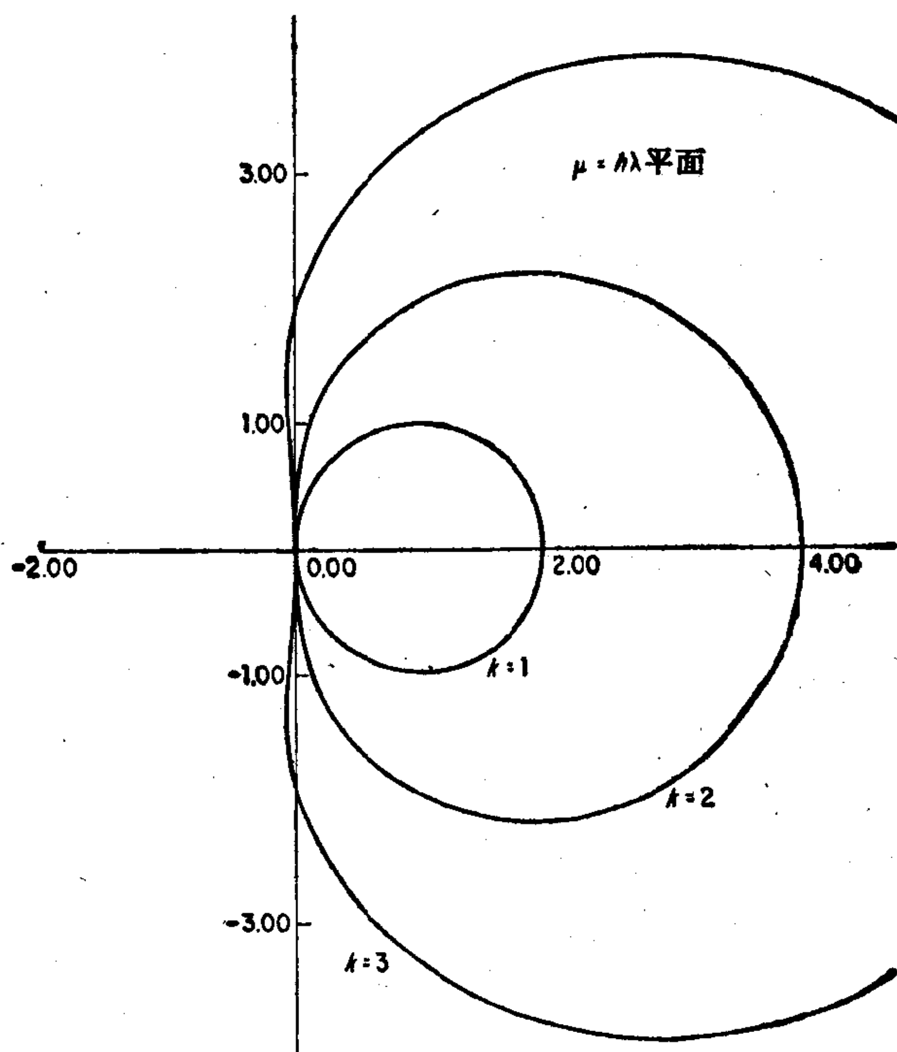


图 11.6. 对一至三阶 Stiff 稳定方法的绝对稳定性区域，方法是在闭合曲线的外部稳定的

以及直到十一阶的多步方法 [Jain (1970)] 均已找到，但可用性尚未检验.

校正方程的求解.

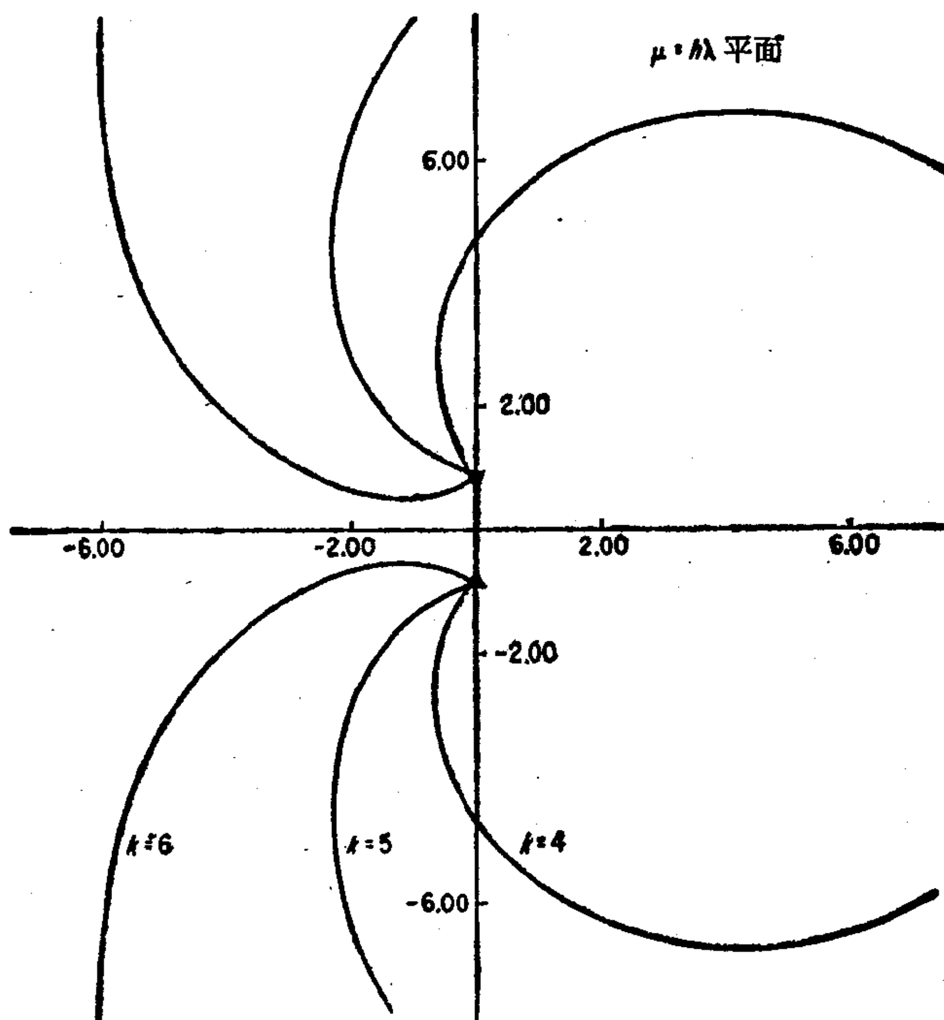


图 11.7. 四至六阶 Stiff 稳定方法的绝对稳定性区域

对于 Stiff 稳定的方法，$\sigma(\xi)$ 至少要有与 $\rho(\xi)$ 相同的阶，否则$\mu=\infty$时一个根为∞. 这表示方法是隐式的，因此，必须解校正方程. 以前的迭代格式为

$$y_n^{(m+1)}=\text{线性和}+h\beta_0 f(y_n^{(m)}).$$

如果

$$\left\|h\beta_0\frac{\partial f}{\partial y}\right\|<1,$$

它是收敛的. 在现在的情形, $h(\partial f/\partial y)$ 可能取相当大的负值,所以,我们必须采用别种迭代格式. 如果 $\partial f/\partial y$ 的计算代价很高,用 Newton 方法求解是很费时的,但是,如果 $\partial f/\partial y$ 变化不是很大,就不需要每次迭代都重新计算. 如果 $\partial f/\partial y$ 不是按每一步均重新计算的 Newton 型方法收敛,则它确实收敛到校正方程的解. 将方法表成 $(k+1)$ 值的标准型,通常做的校正迭代为

$$\mathbf{a}_{n,(m+1)} = \mathbf{a}_{n,(m)} + \mathbf{l}F(\mathbf{a}_{n,(m)}). \tag{11.6}$$

因此,如果方法收敛,它将收敛到

$$\mathbf{a}_n = \mathbf{a}_{n,(0)} + \mathbf{l}\omega, \tag{11.7}$$

其中 ω 为使

$$F(\mathbf{a}_n) = F(\mathbf{a}_{n,(0)} + \mathbf{l}\omega) = 0 \tag{11.8}$$

成立的纯量. 我们用 Newton 迭代求(11.8)的解. 我们将有

$$\omega_{(m+1)} = \omega_{(m)} - \left[\frac{\partial F}{\partial \mathbf{a}} \cdot \mathbf{l}\right]^{-1} F(\mathbf{a}_{n,(0)} + l\omega_{(m)}). \tag{11.9}$$

如果记 $\mathbf{a}_{n,(m)} = \mathbf{a}_{n,(0)} + \mathbf{l}\omega_{(m)}$, (11.9) 变成

$$\mathbf{a}_{n,(m+1)} = \mathbf{a}_{n,(m)} - \mathbf{l}\left[\frac{\partial F}{\partial \mathbf{a}} \cdot \mathbf{l}\right]^{-1} F(\mathbf{a}_{n,(m)}). \tag{11.10}$$

由于 $F(\mathbf{a}) = hf(a_0) - a_1$, 有

$$W = \left[\frac{\partial F}{\partial \mathbf{a}} \cdot \mathbf{l}\right]^{-1} = \left[-l_1 + hl_0\frac{\partial f}{\partial y}\right]^{-1}. \tag{11.11}$$

如果开始多步方法记成

$$y_n = \sum_{i=1}^{k} \alpha_i y_{n-i} + h\beta_0 f_n,$$

有 $l_1 = 1$, $l_0 = \beta_0$. 我们看到, W 依赖于方法的阶(通过 β_0)、h 和 $\partial f/\partial y$. 如果 $\partial f/\partial y$ 是慢变的(在实际中经常发生),则对一步或其中步长和阶不变的若干步,在迭代(11.10)过程中 W 将变化不大.在第 9 章给出的程序中,用了这个事实. 如果要求用 Stiff 方法,就要选择对应于 $\sigma(\xi) = \xi^k (1 \leqslant k \leqslant 6)$ 的 $\mathbf{l}$,

对于这些方法的 α_i, β_0 和 $\mathbf{l}$, 由表 11.1 和 11.2 给出.

表 11.1. Stiff 稳定方法的系数

k	2	3	4	5	6
β_0	$\frac{2}{3}$	$\frac{6}{11}$	$\frac{12}{25}$	$\frac{60}{137}$	$\frac{60}{147}$
α_1	$\frac{4}{3}$	$\frac{18}{11}$	$\frac{48}{25}$	$\frac{300}{137}$	$\frac{360}{147}$
α_2	$-\frac{1}{3}$	$-\frac{9}{11}$	$-\frac{36}{25}$	$-\frac{300}{137}$	$-\frac{450}{147}$
α_3		$\frac{2}{11}$	$\frac{16}{25}$	$\frac{200}{137}$	$\frac{400}{147}$
α_4			$-\frac{3}{25}$	$-\frac{75}{137}$	$-\frac{225}{147}$
α_5				$\frac{12}{137}$	$\frac{72}{147}$
α_6					$-\frac{10}{147}$

表 11.2. 标准型 Stiff 稳定方法的系数

k	2	3	4	5	6
l_0	$\frac{2}{3}$	$\frac{6}{11}$	$\frac{24}{50}$	$\frac{120}{274}$	$\frac{720}{1764}$
l_1	$\frac{3}{3}$	$\frac{11}{11}$	$\frac{50}{50}$	$\frac{274}{274}$	$\frac{1764}{1764}$
l_2	$\frac{1}{3}$	$\frac{6}{11}$	$\frac{35}{50}$	$\frac{225}{274}$	$\frac{1624}{1764}$
l_3		$\frac{1}{11}$	$\frac{10}{50}$	$\frac{85}{274}$	$\frac{735}{1764}$
l_4			$\frac{1}{50}$	$\frac{15}{274}$	$\frac{175}{1764}$
l_5				$\frac{1}{274}$	$\frac{21}{1764}$
l_6					$\frac{1}{1764}$

只当改变阶或者校正 $WF(\mathbf{a}_{n,(m)})$ 在第三次迭代不小的意义下校正过程不收敛时,矩阵 W 才重新计算.

试验例子.

Krogh 非正式提出了检验解 Stiff 方程的程序的例子.

定义

$$(z^i)' = -\beta_i z^i + (z^i)^2, \qquad i = 1, 2, 3, 4,$$

其中 β_i 是非零常数，解为

$$z^i = \frac{\beta_i}{1 + c_i e^{\beta_i t}}. \tag{11.12}$$

如果初始值为 $z^i(0) = -1$, $c_i = -(1 + \beta_i)$，定义酉矩阵 U 为

$$U = \frac{1}{2}\begin{bmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix},$$

并定义 $\mathbf{y} = U\mathbf{z}$，其中 $\mathbf{z} = [z^1, z^2, z^3, z^4]^T$. $\mathbf{y}$ 的微分方程为

$$\mathbf{y}' = -B\mathbf{y} + U\mathbf{w}, \tag{11.13}$$

其中 $B = U\text{diag}[\beta_1, \beta_2, \beta_3, \beta_4]U$, $\mathbf{w} = [(z^1)^2, (z^2)^2, (z^3)^2, (z^4)^2]^T$. 解由 (11.12) 和 $\mathbf{y} = U\mathbf{z}$ 给出. (11.13) 的 Jacobi 矩阵 $\partial f/\partial y$ 为

$$J = U\text{diag}[-\beta_i + 2z^i]U.$$

因此，由于 $U^{-1}=U$，特征值为 $2z^i - \beta_i$. 用初值 $\mathbf{y}(0) = [-1, -1, -1 - 1]^T = \mathbf{z}(0)$，从 (11.12) 看到

$$z^i \to \begin{cases} \beta_i & \text{如果 } \beta_i < 0 \\ 0 & \text{如果 } \beta_i > 0 \end{cases} \text{当 } t \to \infty.$$

因此，特征值 $\longrightarrow -|\beta_i|$. 只要 $\beta_i > 0$ 和 $c_i < -1$，或者 $\beta_i < 0$ 和 $c_i > -1$, z^i 是有限的并且是负的.

根据 Krogh 的建议，对 $\beta_1 = 1000$, $\beta_2 = 800$, $\beta_3 = -10$ 和 $\beta_4 = 0.001$ 的问题积分了.开始，特征值是 -1002, -802.8 和 -2.001. 当 $0.001t \gg 1$ 时，它们是 -1000, -800, -10 和 -0.001. 在积分开始几步，由于含 e^{-1002t} 和 e^{-800t} 项中的

截断误差，步长要加以限制．到时刻 $t=0.01$，这些项就不起作用了．但是，如果用限制步长来保证方法的稳定性，h 必须小于 10^{-3}，而在大部分负半平面上为绝对稳定的方法，允许 h 增大(对这个例子，沿负实轴的稳定性是充分的)．

Stiff 方法的有效性在表 11.3 和 11.4 中表示出来．表 11.3 给出 $t=10^i(i=-2,-1,\cdots,3)$ 后第一个节点上 $\mathbf{y}$ 分量上的最大误差，接着列出了步数，求导数值及矩阵求逆的次数．在程序中允许误差为 10^{-6}，也列出了所用的平均步长，虽然它给出了表 11.3 中步长的一个不理想的估计．可以看到，最后 31 步的平均步长接近30．事实上，t 每增加一个数量级，步长大约增加 10 倍．表 11.4 列出对同一个问题应用 Adams 方法的情形．由于稳定性，步长不能增加，所以到 $t=10$，它大约取 Stiff 方法步数的 80 倍．积到 $t=1000$，大约要取 1.5×10^6步，但是程序当导数求值次数在 $t=16.8$ 超过 10^5 后停止计算．

但是，应该注意，对于 $t<0.01$，当 Stiff 性不成问题时，

表 11.3.　Stiff 稳定方法——Stiff 问题的试验结果

出现的误差	步数	求值数	求逆数	平均步长	当时时间
0.9100D-07	70	179	7	0.1463D-03	0.0102436701
0.2667D-05	110	262	12	0.9535D-03	0.1048869068
0.2208D-05	168	405	15	0.6025D-02	1.0122667324
0.2870D-05	216	523	20	0.4635D-01	10.0110785897
0.2984D-05	252	616	25	0.4067D 00	102.4771283917
0.1199D-05	283	693	29	0.3625D 01	1025.7769259724

表 11.4.　Adams 方法——Stiff 问题的试验结果

出现的误差	步数	求值数	求逆数	平均步长	当时时间
0.1863D-07	60	178	0	0.1698D-03	0.0101889615
0.4525D-06	182	548	0	0.5519D-03	0.1004531167
0.3001D-07	1796	5421	0	0.5569D-03	1.0001087924
0.4639D-05	15285	59246	0	0.6543D-03	10.0004462820

Adams 方法较有利，它用的步数较少，而且比 Stiff 方法更加精确. [这些 k 阶 Stiff 方法的截断误差与 k 阶 Adams-Moulton 方法的截断误差相比为 $\frac{1}{k}$ 与 γ_k^*（见表 7.4）之比.]

图11.8画出积到 $t=100$ 时 $\mathbf{y}$ 的分量的最大误差相对于导数的函数求值数加上偏导数求值次数（它是矩阵求逆数的四倍)的图形(其中离开线的那点，大概是不同分量中误差的巧合而相互抵消的结果). 这儿的结果是用 9.3 节给出的程序得到的,其中参数 $MF=2$.

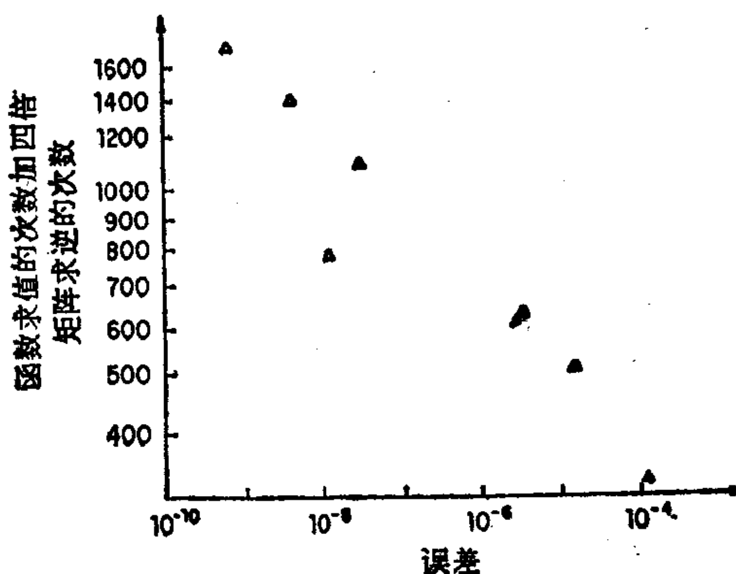


图 11.8. 误差与函数求值的关系

读者注意,这个试验例子不是大范围适定的;接近稳定点时，超过 10^{-3} 的误差可能使扰动解无界.

11.1.2. *A*稳定方法

由于多步方法当其阶大于 2 时不是 A 稳定的，我们自然要问，是否存在 A 稳定的方法类.这一节考察一些方法类，它们不需要关于 $\partial f/\partial y$ 的知识(但要解隐式方程). 对于已提出的需要$\partial f/\partial y$的值或者对其作适当逼近的特殊方法，我们在下

一节讨论.

一类 A 稳定的方法在第 2 章就引进了. Ehle (1968) 指出, $2q$ 阶 q 级隐式 Runge-Kutta 方法是 A 稳定的. 将 $y' = \lambda y$ 代入方程 (2.23), 对 q 级方法有

$$y_{n+1} = \frac{P_q(h\lambda)}{Q_q(h\lambda)} y_n,$$

其中 $P_q(\mu)$ 和 $Q_q(\mu)$ 为 μ 的 q 次多项式. 由于方法是 $2q$ 阶的,

$$R_q(h\lambda) = \frac{P_q(h\lambda)}{Q_q(h\lambda)} = e^{h\lambda} + O(h^{2q+1}).$$

因此, $R_q(\mu)$ 为 e^{μ} 的第 n 对角线 Pade 近似, 并且由 Birkhoff 和 Varga (1965) 知道, 如果 $\mathrm{Re}(\mu) < 0$, 有

$$|R_q(\mu)| < 1.$$

所以, 方法是稳定的.

Axelsson (1969) 定义方法为 Stiff A 稳定的, 如果对方程 $y' = \lambda y$, 当 $\mathrm{Re}(h\lambda) \to -\infty$ 时 $y_{n-1}/y_n \to 0$. 这样的方法使迅速衰减的分量在数值近似中也迅速衰减, 所以, 不需要小的步长, 即使当这些分量仍存在时也是这样 (至少在线性系统中). q 级 $2q$ 阶 Runge-Kutta 方法不是 Stiff A 稳定的, 因为当 $\mu \to -\infty$ 时 $R_q(\mu) \to (-1)^q$. 如果将其阶减小 1, 并且系数限制成使最后计算的 k_i 是区间终点上导数的估计, 则 $e^{h\lambda}$ 由

$$e^{h\lambda} = \frac{P_{q-1}(h\lambda)}{Q_q(h\lambda)} + O(h^{2q})$$

近似. 这个近似被 Axelsson (1969) 证明为 Stiff A 稳定的.

另外一类 A 稳定的方法, 由 Taylor 级数方法的推广给出. 如果能够计算 t_n 和 t_{n+1} 上的导数, 就可以考虑

$$y_{n+1} = y_n + \sum_{j=1}^{q} h^q(\beta_{qj1} y_n^{(j)} + \beta_{qj0} y_{n+1}^{(j)}) \qquad (11.14)$$

型的公式; 如果选取 β_{qji}, 使得与 y_{n+1} 的 Taylor 级的数前 $2q +$

1 项一致，我们找到

$$\beta_{qj} = \beta_{qj1} = (-1)^{j+1}\beta_{qj0}.$$

对于 $q = 1$，我们得到梯形法则 $\beta_{11} = \frac{1}{2}$；对于 $q = 2$，得到

$$y_{n+1} = y_n + \frac{h}{2}(y'_n + y'_{n+1}) + \frac{h^2}{12}(y''_n - y''_{n+1}).$$

如果 (11.14) 被用来积分 $y' = \lambda y$，我们得到

$$y_{n+1} = \frac{1 + \sum_{j=1}^{q} \beta_{qj1}(h\lambda)^j}{1 - \sum_{j=1}^{q} \beta_{qj0}(h\lambda)^j} y_n = R_q(h\lambda)y_n.$$

由于阶为 $2q$，$R_q(h\lambda)$ 又是 $e^{h\lambda}$ 的对角线 Pade 近似，所以是 A 稳定的．将其阶减小 1，并令 $\beta_{qq1} = 0$，我们得到 Stiff A 稳定方法．

由于解隐式方程 (2.23) 或者计算导数及解隐式方程 (11.14) 均要作大量的工作，因此这些方法除低阶的情形外，没有一个得到广泛的应用，低阶的情形为 Taylor 级数方法的梯形法则和 Runge-Kutta 方法的隐式中点法则．后者由

$$k_1 = hf\left(y_n + \frac{k_1}{2}\right),$$

$$y_{n+1} = y_n + k_1$$

给出．

11.1.3. 基于 $\partial f/\partial y$ 的知识的方法

直接应用 $A = \partial f/\partial y$ 的各种方法已经提出了．如果 A 的特征值已知，则可推出对具有那些衰减速度的指数函数为精确的方法，就象我们选取多步方法对多项式为精确一样 [Pope (1963)]．文献中提出的一些方法，要求特征值知识或 A 的近似的知识 [Lawson (1967) 和 Dahlquist (1969)]．

线性方程.

设有方程组 $\mathbf{y}' = A\mathbf{y}$,

其中 A 是常矩阵，并且可以找到相似变换，使得 QAQ^{-1} 为 Jordan 型. 于是,我们能够直接解方程

$$\mathbf{z}' = J\mathbf{z},$$

其中 $\mathbf{z} = Q\mathbf{y}$, $J = QAQ^{-1}$. 如果作变换

$$\mathbf{z}(t) = e^{Dt}\boldsymbol{\eta}(t),$$

这里 D 是 J 的对角线部分, e^{Dt} 为对角线矩阵,对角线上的每个元为对应于 Dt 的对角线元素的指数,我们得到方程

$$\boldsymbol{\eta}'(t) = e^{-Dt}(J - D)e^{Dt}\boldsymbol{\eta}(t) = (J - D)\boldsymbol{\eta}(t).$$

由于矩阵 $J - D$ 的上对角线元素最大为 1, $\boldsymbol{\eta}(t)$ 是多项式,我们得到

$$\mathbf{y} = Q^{-1}\mathbf{z} = Q^{-1}e^{Dt}\boldsymbol{\eta}(t).$$

一般, A 不是固定的,但如果对给定的 $A(t_0)$ 值作上述变换,得到的微分方程虽然非常复杂,可是因为特征值很小,可用通常的方法来处理,而没有稳定性问题. 但这种方法需要找到矩阵 Q, 这需要大量的工作. 如果 $A(t)$ 的变化很快,则新的 Q 经常要重新计算,使得此方法由于工作量过高而不能用.

Runge-Kutta 方法的推广.

Rosonbrock (1963) 提出了一个显式 Runge-Kutta 过程的扩充,使其含有 $\partial f/\partial y$. 最一般的形式如下:

$$k_1 = hf(y_n) + hb_1A(y_n)k_1,$$

$$k_2 = hf(y_n + \beta_{21}k_1) + hb_2A(y_n + \eta_{21}k_1)k_2,$$

$$\cdots\cdots\cdots\cdots\cdots\cdots$$

$$k_q = hf\left(y_n + \sum_{i=1}^{q-1}\beta_{qi}k_i\right) + hb_qA\left(y_n + \sum_{i=1}^{q-1}\eta_{qi}k_i\right)k_q,$$

$$y_{n+1} = y_n + \sum_{i=1}^{q} \gamma_i k_i, \tag{11.15}$$

其中 $A(y) = \partial f(y)/\partial y$. 这些方法在文献中有许多特殊的例子 [Calahan (1968) 和 Allen (1969)]. 由于它们用到 $\partial f/\partial y$, 在某种意义下, 它们是隐式的方法. 对问题 $y' = A(t)y$ 用隐式多步方法或 Runge-Kutta 方法必须要解的方程的形式类似于(11.15). 这个方法的一个例子为 Calahan (1968) 的三阶方法, 由

$$\begin{aligned}
k_1 &= hf(y_n) + hb_1 A(y_n) k_1, \\
k_2 &= hf(y_n + \beta_{21} k_1) + hb_1 A(y_n) k_2, \\
y_{n+1} &= y_n + \gamma_1 k_1 + \gamma_2 k_2
\end{aligned} \tag{11.16}$$

给出, 其中

$$b_1 = \frac{1}{2}\left(1 + \sqrt{\frac{1}{3}}\right) = 0.788675,$$

$$\beta_{21} = -2\sqrt{\frac{1}{3}} = -1.154701,$$

$$\gamma_1 = 0.75, \quad \gamma_2 = 0.25.$$

当这类方法应用于方程 $y' = \lambda y$ 时, 导出

$$y_{n+1} = R(h\lambda) y_n$$

形的关系式, 其中 $R(h\lambda)$ 为 $h\lambda$ 有理多项式. 如果对于 Re $(h\lambda) < 0$ 能够证明 $|R(h\lambda)| \leqslant 1$, 则方法是 A 稳定的. 为了证明这一点, 只要证明当 $\mu \to \infty$ 时, $\lim |R(\mu)| \leqslant 1$, $|R(i\omega)| \leqslant 1$, $-\infty < \omega < \infty$, 并在一些点上不等号成立, 以及在 $\mathrm{Re}(\mu) \leqslant 0$ 中, $R(\mu)$ 是正则的.

11.2. 代数方程和奇异方程

有两类与 Stiff 方程有关的问题. 第一类称为奇异摄动. 在这个问题中, 方程组对两个向量 $\mathbf{y}$ 和 $\mathbf{z}$ (不一定维数相同)

给定形式为

$$\mathbf{y}' = f(\mathbf{y}, \mathbf{z}, t), \tag{11.17a}$$

$$\varepsilon\mathbf{z}' = g(\mathbf{y}, \mathbf{z}, t), \tag{11.17b}$$

其中 ε 为非常小的常数．在很小的区间 $[0, t_\varepsilon]$ 内，由于(11.17b)，解显示出迅速变化的性态(这问题常常称作边界层问题，由于它出现在接近于边界时的流体动力流问题中)；然后第二个方程可成功地用

$$0 = g(\mathbf{y}, \mathbf{z}, t) \tag{11.18}$$

代替．

当高阶隐式方程

$$F(y, y', \cdots, y^{(p)}, t) = 0 \tag{11.19}$$

在 t 增加时，$\partial F/\partial y^{(p)} \to 0$ (或变得很小)，出现第二个问题．显然，当 $\partial F/\partial y^{(p)} = 0$ 时，(11.19) 变为阶为 $p-1$ 或更低阶的方程．例如，考虑线性二阶方程

$$\varepsilon(t)y'' + ay' + by = g(t),$$

其中当 t 增加时，$\varepsilon(t) \to 0$．如果记 $z = y'$，我们得到方程组

$$y' = z, \tag{11.20a}$$

$$\varepsilon(t)z' = -az - by + g(t). \tag{11.20b}$$

如果这些方程由于小的 ε 使得暂态过程衰减掉，将存在一个边界层，接着为 (11.20b) 可用

$$0 = -az - by + g(t)$$

代替的区域．将其代入 (11.20a)，我们得到一阶方程

$$y' = -\frac{by - g(t)}{a}.$$

在另外一类问题中，开始给出微分方程 (11.17a) 和代数方程 (11.18) 的组．我们看到，在所有三种情形中，都存在这个问题的求解区域．在前两种情形，都存在这样的区域，在其中必须首先解全部微分方程问题，然后解决确定何时转换成

另一种方法的附加问题.

这一节我们考察一种方法，这种方法使得这过程的二步可以统一处理. 这样，就没有必要直接对 $y^{(p)}$ 或 $y^{(p-1)}$ 解(11.19)或者决定那个区域是否适当.

考虑解(11.19)的多值方法的标准形式. 欲确定 ω,使

$$\mathbf{a}_n = A\mathbf{a}_{n-1} + \mathbf{I}\omega$$

满足(11.19). 令 $F(\mathbf{a}) = F(a_0, a_1/h, 2!a_2/h^2, \cdots, p!a_q/h^p, t)$. 如果 F 具有连续导数,则存在一个 ξ,使得

$$\mathbf{a}_n = A\mathbf{a}_{n-1} - \mathbf{I}\left[\frac{\partial F}{\partial \mathbf{a}}(\xi) \cdot \mathbf{I}\right]^{-1} F(A\mathbf{a}_{n-1}) \quad (11.21)$$

(在数值上,可用类似于拟 Newton 方法

$$\mathbf{a}_{n,(0)} = A\mathbf{a}_{n-1},$$

$$\mathbf{a}_{n,(m+1)} = \mathbf{a}_{n,(m)} - \mathbf{I}\left[\frac{\partial F}{\partial \mathbf{a}} \cdot \mathbf{I}\right]^{-1} F(\mathbf{a}_{n,(m)})$$

的迭代方法找到这个解，其中 $\partial F/\partial \mathbf{a}$ 象对 Stiff 方法那样在它的变量的某一个适当点上求值). 如果考察(11.21)对数值扰动 $\mathbf{e}_n$ 的稳定性,得到

$$\mathbf{e}_n = S_n\mathbf{e}_{n-1} + O(\mathbf{e}_n + \mathbf{P}_n)^2,$$

其中 $\mathbf{P}_n$ 为预估中截断误差的阶的量(如果 F 对 $\mathbf{a}$ 是线性的，这最后一项就没有了),

$$S_n = \left[I - \left[\frac{\partial F}{\partial \mathbf{a}}(\xi) \cdot \mathbf{I}\right]^{-1} \mathbf{I}\frac{\partial F}{\partial \mathbf{a}}(\xi_1)\right] A,$$

ξ_1为两个数值解中间的一个点. 假定 $\partial F/\partial y^{(p)}$ 不为零，对于小的 h, 我们找到

$$S_n = \left[I - \frac{\mathbf{I}\boldsymbol{\delta}_p^T}{l_p}\right] A + O(h) = S + O(h). \quad (11.22)$$

注意，这与第九章中得到的误差增大矩阵 S 仅相差 $1/l_p$ 项(事实上,表 9.1 中给出对 p 阶方程的 Adams 型方法有 $l_p = 1$, 所以不存在差别).

在第 9 章，我们找到的 $\mathbf{I}$ 依赖于求解方程的阶．如果能找到不依赖于 p 的 $\mathbf{I}$，使得由 (11.22) 给出的 S 对若干个 p 的值稳定，则得到的方法就可用来解隐式方程 (11.19)，而不需要知道它的实际阶，只要知道它的阶使 S 稳定．一个有趣的事实是，表 11.2 中给出的 $\mathbf{I}$ 使得 S 对所有 $0 \leqslant p \leqslant k$ 稳定．因此，这些方法对 $k \leqslant 6$ 的所有 k 阶隐式方程均可应用．如果阶是固定的，即 $\partial F/\partial y^{(p)}$ 距零是有界的，则不管 p 在范围 $0 \leqslant p \leqslant k$ 中为何值，当 $h \to 0$ 时方法收敛．

但是，在区间中某个点上，$\partial F/\partial y^{(p)}$ 趋于零的情形，可能出现问题，这可以考虑方程

$$F(y, y', t) = g(t) + yf(t) - \varepsilon(t)y' = 0 \quad (11.23)$$

看出．对于这个问题，S_n 由

$$S_n = (I - \mathbf{I}[l_0 hf(t_n) - l_1\varepsilon(t_n)]^{-1}[hf(t_n)\boldsymbol{\delta}_0^T - \varepsilon(t_n)\boldsymbol{\delta}_1^T])A$$

给出．如果当 t 增加时 $\varepsilon(t)$ 趋于零，但 $f(t) \neq 0$，我们可在范围 $0 < h \leqslant h_0$ 中找到一个 h，使得对一些 t，$\mu = hf(t)/\varepsilon(t)$ 取负半平面中的任意值(正半平面中的点勿需考虑，因当 t 增加时，微分方程愈来愈不稳定)．因此，必然涉及当 $\mathrm{Re}(\mu) \leqslant 0$ 时

$$S_n = \left[\frac{I - \mathbf{I}[\mu\boldsymbol{\delta}_0^T - \boldsymbol{\delta}_1^T]}{l_0\mu - l_1}\right] A$$

的稳定性．这恰好是上一节所讨论的 Stiff 稳定性问题，而且我们已经看到，在整个 $\mathrm{Re}(\mu) \leqslant 0$ 中可以对二阶方法使 S_n 是稳定的，在 $\mathrm{Re}(\mu) \leqslant 0$ 的大部分可由至少直到十一阶的方法使 S_n 稳定．对 p 阶方程，我们注意

$$S_n = \left[\frac{I - \mathbf{I}\left[\sum_{i=0}^{p} \mu_i \boldsymbol{\delta}_i^T\right]}{\sum_{i=0}^{p} \mu_i l_i}\right] A$$

对任何稳定的形如

$$\mu_0 y + \mu_1 h y' + \cdots + \mu_p \frac{h^p y^{(p)}}{p!} = 0$$

的方程的稳定性区域. 但到目前为止还不知有任何结果.

代数方程.

当(11.19)中的 $p = 0$ 时,我们求解一个代数方程. 在这种意义下, 将代数方程看成是阶为零的微分方程. 如果用上面提出的方法来解那样的方程, 我们得到由 (11.22) 给出的 $p = 0$ 的误差增大矩阵 S. 我们已经指出, 由表 11.2 给出的 **I** 对 Stiff方程是稳定的.

事实上,在这种情形, S 的所有特征值均为零. 这可由观察方程

$$\varepsilon y' = f(y, t)$$

当 $\varepsilon \to +0$ 的极限看出. 如果 $\partial f/\partial y < 0$, 对小的正 ε,这是一个 Stiff 方程. S 的特征值是 $\rho(\xi) + (1/\varepsilon)(\partial f/\partial y)\sigma(\xi)$ 的零点. 当 $\varepsilon \to 0$ 时,它们变成 $\sigma(\xi)$ 的零点,由于 $\sigma(\xi) = \xi^k$,它们都是零. 当 $\varepsilon = 0$ 时,微分方程变成代数方程.

当这个方法应用于代数方程时, 我们用前面的 t_i 上的 y 值作多项式外插来进行预估 $F(y, t) = 0$ 在 t_n 的解, 然后用解 (10.19) 的方法进行校正. 如果应用 Newton 校正迭代,它与用 Newton 方法解方程 $F(y, t) = 0$ 是一样的.

直接利用过 $y_{n-1}, y_{n-2}, \cdots, y_{n-k-1}$ 的外插公式寻找 y_n 的初始近似,然后用 Newton 方法,也可以得到这种类型的方法. 如果校正迭代到收敛,没有一个误差的传播会超过 $k + 1$ 步, 这种方法的误差增大矩阵的特征值均为零. 因此, 这种方法等价于前面的方法,并且将其写成标准形式时具有相同的 **I**.

一种方法可以同时求解代数方程和微分方程这一点, 对在网络分析和模拟中出现的方程类型是重要的. 它们常常是

形如

$$\mathbf{F}(\mathbf{y}, \mathbf{y}', t) = 0$$

的隐式方程组，其中 $\mathbf{F}$ 和 $\mathbf{y}$ 都是向量，它们表示 s 个方程和 s 个因变量. 在这种情形,每个因变量的校正是

$$\left[\frac{\partial \mathbf{F}}{\partial \mathbf{y}} \mathbf{I}_0 + \frac{\partial \mathbf{F}}{\partial \mathbf{y}'} \frac{\mathbf{I}_1}{h}\right]^{-1} \mathbf{F}(\mathbf{y}, \mathbf{y}', t)$$

的分量,没有必要直接对 $\mathbf{F}$ 解出 $\mathbf{y}'$ 或者确定哪个方程是微分方程. 矩阵$\left[\frac{\partial \mathbf{F}}{\partial \mathbf{y}} \mathbf{I}_0 + \frac{\partial \mathbf{F}}{\partial \mathbf{y}'} \frac{I_1}{h}\right]$通常是稀疏矩阵,在这种公式中用稀疏的技巧是合适的. [见 Tewardson (1967), Tinney 和 Walker (1967) 和 Willoughby (1969)]

如果公式中对一些因变量呈线性关系，并且不出现导数(这是经常发生的),我们将这些因变量重记成 $\mathbf{v}$, 方程记成

$$\mathbf{F}(\mathbf{y}, \mathbf{y}', t) + P\mathbf{v} = 0,$$

其中 $\mathbf{v}$ 的分量已经从 $\mathbf{y}$ 中去掉, 并且 P 为常矩阵. 于是可以看到,在校正迭代中 $\mathbf{y}$ 的变化与 $\mathbf{v}$ 的预估值无关. 因此,仅须贮存 $\mathbf{v}$ 的值，而不要它的导数，并且对 $\mathbf{v}$ 不需要应用预估过程. 这种处理的更详细的情形可以在 Gear (1971) Calahan (1969) 和 Hachtel 等 (1971) 中找到.

11.3. 参数估计

一般,常常假定系统的性能由依赖于参数 $\mathbf{P} = \{p^i\}$ 的微分方程组给出. 在不同时刻的实验量测值给出解的近似值,问题是要寻找这些参数. 假定方程组为

$$\mathbf{y}' = \mathbf{f}(\mathbf{y}, t, \mathbf{P}) \qquad \mathbf{y}(0) = \mathbf{y}_0(\mathbf{P}) \qquad (11.24)$$

并且给定 $\mathbf{z}_i = \mathbf{y}(\tau_i)$ $(i = 1, \cdots, m)$. 如果刚好给出与参数和方程个数相同的 $\mathbf{y}(\tau_i)$ 的值，我们要解一种类型的边值问题. 通常 $\mathbf{y}(\tau_i)$的值比参数及初值能够满足的数目要多,所以,

用最小二乘方的处理来满足这些条件是合适的.(11.24)的解可表成

$$\mathbf{y}(t) = F(t, \mathbf{P}), \tag{11.25}$$

最小平方拟合要求形如

$$\sum_{i=1}^{m} \|\mathbf{y}(\tau_i) - \mathbf{z}_i\|_i^2 \omega_i^2 \tag{11.26}$$

的函数达到极小值，其中 $\|\cdot\|_i$ 是可以排除向量的一些分量(如果它们没有被指定的话)的模.这里不讨论求极小值问题，读者可以参考这一领域中的许多文章和书，例如 Kowalik 和 Osborne (1968).

一些方法需要计算(11.26)对参数的偏导数,这可以数值地用差分来做到,如果有 q 个参数及 $\mathbf{y}$ 有 s 个分量,则需要对 s 个方程积分 $q+1$ 次. 另外,我们可以得到 $\partial y^i/\partial p^j$ 的微分方程. 从(11.24),有

$$\frac{d}{dt}\left(\frac{\partial \mathbf{y}}{\partial \mathbf{P}}\right) = \frac{\partial \mathbf{f}}{\partial \mathbf{P}} + \frac{\partial \mathbf{f}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{P}}. \tag{11.27}$$

(11.24)和(11.27)是必须积分一次的$(q+1)s$ 方程组.如果 $\partial \mathbf{f}/\partial \mathbf{P}$ 很复杂，这种处理似乎比数值微分费事,但是，另外一种直接微分好处极大. 特别,如果原始问题是 Stiff 的，矩阵 $\partial \mathbf{f}/\partial \mathbf{y}$ 必须用某种方式近似. 因为

$$\frac{\partial\left[\frac{d}{dt}\left(\frac{\partial \mathbf{y}}{\partial p^j}\right)\right]}{\partial \frac{\partial \mathbf{y}}{\partial p^j}} = \frac{\partial \mathbf{f}}{\partial \mathbf{y}},$$

所以两者有相同的特征值,于是组(11.27)也是Stiff的. 可以将校正先应用到(11.24)来计算 $\mathbf{y}_{n+1}$, 然后处理(11.27). 对于每个 p^j,在校正的 Newton 迭代中,向量 $\partial \mathbf{y}/\partial p^j$ 使用了 $\mathbf{y}$ 所使用的相同矩阵,所以它不需要重新计算.

如果初始值未知，它们可以考虑成参数. 如果测量 $\mathbf{z}_i$ 的

时刻 τ_i 中有误差，由于

$$\left.\frac{\partial \mathbf{y}(\tau_i)}{\partial \tau_i}\right|_{P\text{固定}} = \mathbf{f}(\mathbf{y}(\tau_i), \tau_i, \mathbf{P})$$

已知，它们可以包含在使(11.26)达到极小值的未知量中.

问　　题

1. 证明：在11.1.1节的试验例子，当系统达到稳定后，如果 y 在每个分量上的扰动超过0.005(带适当的符号)，不是适定的.
2. (a) 应用方法

$$q_1 = y_n + \beta_{11}hf(q_1) + \beta_{12}hf(y_{n+1}),$$
$$y_{n+1} = y_n + \beta_{21}hf(q_1) + \beta_{22}hf(y_{n+1}),$$

对方程 $y' = \lambda y$，用 y_n, λ 和 h 表达 y_{n+1}.

(b) 这个方法的最高阶是多少?

(c) 对这个阶 β_{ij} 的值如何?

3. 假定应用梯形法则从 t_n 到 $t_n + h$ 积分一步，得到 $\tilde{y}_{n+1}$，然后用 $h/2$ 步长积分二步得到 $\hat{y}_{n+1}$，可以证明

$$y_{n+1} = \frac{4\hat{y}_{n+1} - \tilde{y}_{n+1}}{3}$$

的局部截断误差是 $0(h^5)$. 这方法 A 稳定吗?

4. 证明由(11.16)给出的方法是 A 稳定的.
5. 如果用 Euler 方法积分方程

$$y' = 10(e^t - y) + e^t, \quad y(0) = 1,$$

问到什么精确度时 Stiff 性成问题了?

6. 考虑单个的一阶方程

$$y' = f(y,t), \quad y(0) = y_0, \quad t \in [0,b],$$

其中 f 有连续的有界的导数，并且对这个特殊的问题有

$$\left|\frac{\partial f}{\partial y} - L\right| < u,$$

其中 L 和 u 均是常数，使得 $0 < u < -L$. 用固定步长的多值方法积分这个问题，并且校正是用 Newton 方法迭代到收敛；知道它的局

部截断误差对 $h \leqslant h_0$ 以 Th^{r+1} 为界以及矩阵

$$S(h)=\left[\frac{l-\mathbf{1}(\delta_1^T-hL\delta_0^T)}{l_1-hLl_0}\right]A$$

对 $\delta \leqslant h \leqslant h_0$ 所有特征值均小于1；还知道这方法是稳定的[$S(0)$满足根条件]. 对 $\delta \leqslant h \leqslant h_0$ 推导与 L 无关的误差的界.

7. 已知化学实验中二种成份的浓度满足常微分方程组

$$\frac{dy}{dt}=-k_1y+k_2(b-2y-z)z,$$

$$\frac{dz}{dt}=-k_3z+k_4(b-2y-z)(a-y-z)-\frac{dy}{dt},$$

其中 k_i 是未知的正常数，而 a 和 b 是已知的正常数.

作了下面的试验：

(a) 当 $t=0$ 时，令 $a, b, y(0)$ 和 $z(0)$ 的值为

$$a=1.0,\ b=2.0,\ y(0)=0.25,\ z(0)=0.50.$$

(b) 在不同时刻 $y(t)$ 和 $z(t)$ 的样本值，数据如下：

t	$y(t)$	$z(t)$
0	0.250	0.500
0.333	0.301	0.403
0.672	0.324	0.362
1.012	0.335	0.345
∞	0.345	0.332

化学家认为 $t=\infty$ 表示使组达到稳定 $\left(\text{即}\ \frac{dy}{dt}=\frac{dz}{dt}=0\right)$ 的充分长的时间. 在这种情形，$t=100$ 组一定稳定了.

从直观考虑，化学家知道 k_i 应该接近于1，你能为他计算出 K_i 的什么值？估计这些解中的误差.

8. 证明 11.2 节最后的 $\mathbf{v}$ 的预估值不影响校正迭代的命题是正确的(忽略舍入误差).

12. 方法的选取

这一章给出(对给定问题)选取方法的一些规则．这些说明是根据方法的目前状态给出的，因此，还将讨论一些问题，这些问题的进一步发展可能改变这些规则．

对于给定的方法类，选取步长和阶的准则在第 5 章对单步方法讨论过了．对其它方法，如果仍假定目的是为了得到每一次函数求值可达到最大的步长，则这些准则同样适用．选取等价的多步方法的不同表达式在第 9 章讨论了．因此，还需要在各种类型的方法(Taylor 级数，Runge-Kutta，多值，Bulirsch-Stoer 以及用到方程的更多知识的一些方法)中进行选择，而且在 Runge-Kutta 和多值方法的情形，要在方法的常数可以取的各种值中进行选取(这些值影响 Runge-Kutta 方法的截断误差，而在多值方法的情形，影响截断误差和稳定性区域)．

普通的问题．

可以将在数字计算机上求解的许多方程归入普通问题类，因为即使用最差的求解方法，耗费的机器时间也很少．这时决定最好方法的原则是要尽量减少人准备程序的时间．于是具有定步长的古典 Runge-Kutta 方法是最方便的一个方法．如果欲确定精度是否满足要求，可将步长缩小 $\frac{1}{2}$．但是，如果计算机程序库中包含写成标准程序而不需要特殊起始过程的其它方法，则它们也一样可以用．如果问题简单(例如 y 是线性的)，又是 Stiff 的(象用 Runge-Kutta 方法，需要非常小的步长)，于是用梯形法则有效．但是，必须解一个隐式方程．在线

性方程的情形，可以用直接法求解. 子程序库中通常有这样的方法可用.

光滑的非 Stiff 问题.

无论 Bulirsch-Stoer 或者多值方法，对这些问题都是好的. 对于许多问题，Bulirsch-Stoer 方法属于最快的方法之列，但是直到现在，还不知道确定最好的阶和步长的准则. 象第 9 章中给出的自动方法那样的多值方法在起始计算中要花费相当的时间，但在以后就变得十分有效. 因此，如果微分方程要在大区间(在为了精度要求需要的步数多少的意义下)上进行积分以及导数的相对大小变化不快，多值方法可能是最好的. 如果知道问题不具有接近增长分量的增长速度衰减的任何分量，则对于弱稳定方法或与其相近的方法，可能的较小截断误差是值得考虑的. 否则，Adams 方法是更好的.

具有间断的问题.

不少问题在许多点上有间断的导数. 例如，火箭轨道当发动机开或关时，将具有间断的二阶导数. 在这样的点上采用 Taylor 级数展开的方法，必须对方法的阶加以限制. 例如对火箭轨道，在出现间断的步上不能用大于一阶的方法. 只要间断在区间 $[t_{n-k}, t_n]$ 的内部，k 步方法就应该限制成一阶方法. 在这种情形，步长使得间断点仅出现在节点上的单步方法是最好的，因为它的阶没有受到限制. 对这些问题，无论 Runge-Kutta 方法还是 Bulirsch-Stoer 方法都是合乎需要的.

高阶方程.

高阶方程直接方法的试验 [Gear (1967)] 和初步的结果 [Rutishauser (1960)]表明，在某些情形，将方程转换成较低阶的方程组是较好的，而在另外一些情形，最好直接处理高阶方程. 如果用后一种方式，则其中 Adams 方法的多值推广是最方便的，因为高阶方程的程序本质上等同于第 9 章中给出

的一阶程序.

Stiff 方程.

对于 Stiff 型一般非线性方程，可用第11章讨论过而在第 9 章编成程序的多步方法来处理. 其它的方法［Allen (1969), Calahan (1968), Pope (1963), 和 Rosenbrock(1963)］在某些情形可能会更好一点，但将它们写成处理许多情形的一般程序是比较困难的. 隐式 Runge-Kutta 方法看来有希望处理那些可能出现间断的情形，特别是 Axelsson (1969) 的那些方法，它们是 Stiff A 稳定的. 但它们需要相当多的函数求值次数，为了收敛到 q 级隐式方程的解，大概需要比多值方法更多的计算，因为方程组的个数是原来的 q 倍.

如果方程是线性的，或者 Stiff 部分出现在线性项中，则 Lawson (1967) 和 Dahlquist (1969) 的方法也许最合适，虽然需要寻找适当的方法来控制这些方法中的误差.

当 $\partial f/\partial y$ 有接近于虚轴的特征值时，会出现类似于 Stiff 方程的情形. 如果这些特征值比较固定，众所周知，解中具有给定频率的振动分量. Gautschi (1916) 就已经提出对低阶三角多项式为精确的方法，就象通常的多步方法对低阶的普通多项式精确一样.

五种方法的比较.

作为前面各章提出的各种方法的相对有效性的导引，对 6.2 节最后给出的三个问题（负指数，Euler 方程，和 J16）用下面几种方法进行积分：第 5 章的自动 Runge-Kutta 程序 (RK)，第 6 章的多项式和有理外插程序 (POLY 和 RAT) 以及用 Adams 或者 Stiff 公式的多值方法 (ADAM 和 STIFF). 外插算法的参数 MAXORD 和 MAXPTS 分别取 6 和 8，每步的误差为 $O(h^{14})$. ADAM 和 STIFF 的最大阶分别取 7 和 6. 试取的初始步长对 RK, ADAM 和 STIFF 取 EPS，对 POLY 和 RAT 取

(EPS)[25].（后面两个方法对初始步长的选取是很敏感的，这表示步长控制部分需要改进）

实际的相对误差和函数求值次数之间的关系由图12.1到图12.3表出．为达到这一点，在exp(－20)和Euler方程的情形，EPS从10^{-2}变到10^{-11}；在J16的情形，从10^{-6}变到10^{-11}．

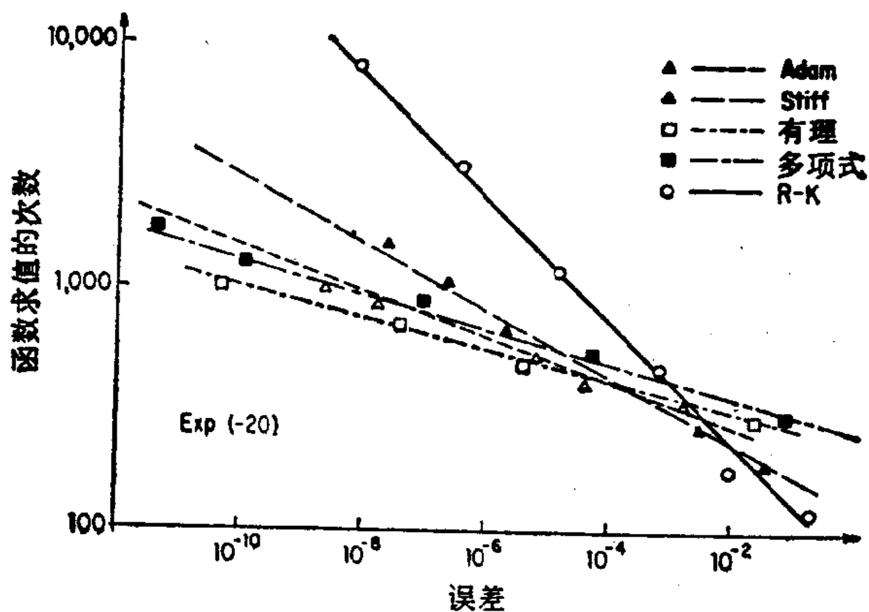


图12.1　在exp(－20)中的误差

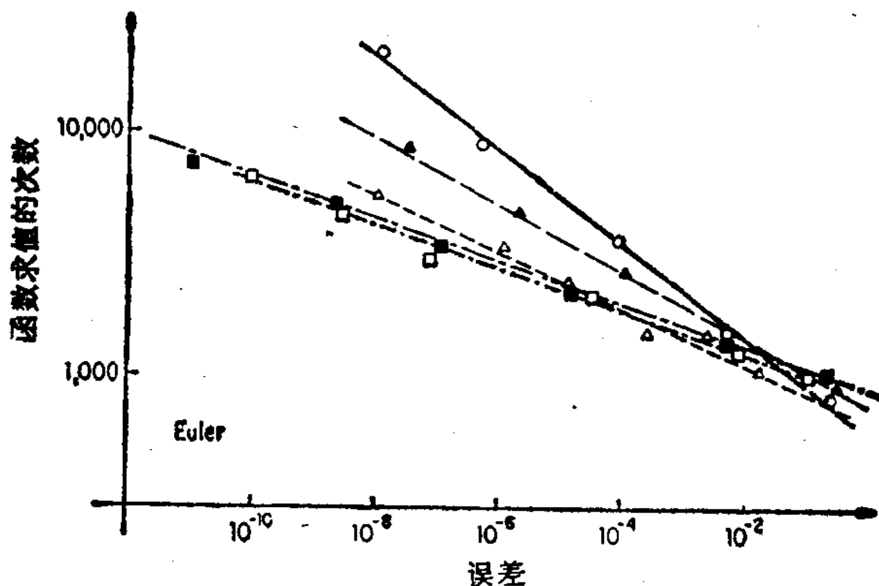


图12.2　Euler方程的误差

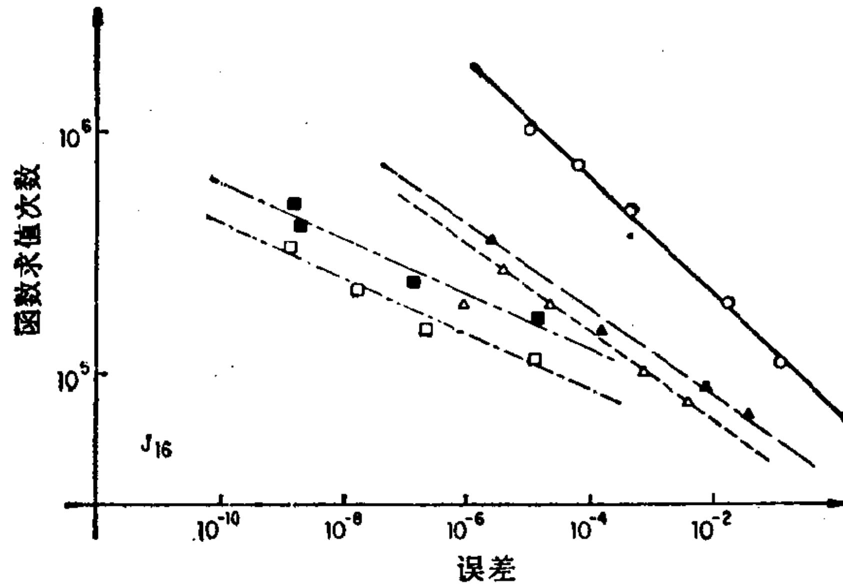


图 12.3　J16 方程中的误差

Runge-Kutta 图形较陡的斜率表明,在精度低时很好. 对这些试验，外插方法一般比多步方法好. 由于外插方法的初始步长必须小心选取,选得不好会比多值方法的结果更坏;以及由于这里只用了七阶的 Adams 方法,所以这一点部分地被抵消. 如果更高的最高阶方法已编成程序，这些光滑的问题将在很少几步就可以积分完.

如果将子程序所用的时间取成 $A+B\times N$ 的形式,其中 N 是方程组中方程的个数，表 12.1 列出了子程序所用的时间(在 360/91 系统上用了 IBM 360 FORTRAN H, OPT = 2). 可以看到，Runge-Kutta 对每次函数求值，执行速度比其它方

表 12.1. 函数求值时 IBM360/91 的执行时间(以10^{-6}秒为单位)

方法	*Adam*	*Stiff*	*Rk*	*Rat*	*Poly*
无方程时的时间	112.7	92.5	49.7	33.3	28.8
每一附加方程的时间	17.9	45.6	1.2	5.8	5.0

法快得多．对于函数求值时间少的问题，程序本身的工作量就很重要．因此，甚至对中等的精度，Runge-Kutta 方法也是合适的．如果函数求值的时间长，只有函数求值的次数应该被考虑．

12.1. 发展概貌

有关这个论题的评论，均属设想，并且在很短的时间内，预期部分地或全部地得以证明是错误的．将它作为最后一节的理由，是想提出一些需要进一步研究的比较明显的领域．这些领域叙述如下：

1. 对形如

(a) $y' = Ay + f(t)$,

(b) $y' = A(t)y + f(t)$

的线性或拟线性方程的方法．

2. 寻求何时应将高阶方程直接处理或将其转换成较低阶方程组的准则．

3. Stiff 方程的外插方法．

4. 建立 Stiff 方法的适当理论．

5. 对变步长的多值方法进行理论推广．

虽然目前已有的方法对于其解或者 $\partial f/\partial y$ 的特征值预先一无所知的一般非线性方程似乎是有效的，但是对范围较小的方程，特别是线性或拟线性方程，总可用比目前的方法更为有效的方法来处理．许多大的系统，例如出现在网络上，含有很大的线性分量，但是除了简单的情形外，这些性质几乎还没有真正探索过．

在实际中并不经常遇到超过二阶的方程．除了缺少二阶导数的情形外，何时使用直接法，是不明显的．由于归结成一阶方程组所使用的信息几乎将处理的信息增加一倍，所以，期

望大多数合适的直接法将速度加快一倍似乎是合理的.

上述关于外插法的试验表明,当对非 Stiff 方程与其它方法比较时，它有一种可能有利的性质. Dahlquist (1969)已经指出，将外插法应用到梯形法则对 Stiff 方程可能是有效的. 外插方法似乎提供了对阶和步长控制的最灵活的处理，并可以代替解一般非线性问题的其它方法.

Stiff 问题的理论工作,进展不大. 最迫切需要的是,不依赖于 Jacobi 矩阵特征值的负部的误差界,以及欲达到这样的结果系统要满足的最小附加条件概念. Dahlquist (1969) 对非线性方程讨论了梯形方法.

许多方法还没有进行讨论,对此,作者相信它们最近还不会超过本书强调的三种主要方法中最好的方法. 但是，在这些方法中,有许多还没有象 Runge-Kutta,Adams 或者外插方法那样得到普遍的注意,必须承认,也许在另外的方向上会取得很大的进展.前面没有讨论的方法中有 Runge-Kutta 方法与多步方法的联用 [Butcher(1965B),Gragg 和 Stetter(1964),和 Gear (1965)],其中用到由前面的点得到的信息和积分区间中非节点上由函数求值得到的信息. 作者的初步试验表明，它们的绝对稳定性区域是不大的,并且考虑到误差估计和步长选取,比起简单的 Adams 方法要求更多的函数求值次数.

参考书目

Allen, B. T. (1960), "A New Method of Solving Second Order Differential Equations When the First Derivative is Present," *Comp. J.*, **8**, pp. 392–394.

Allen, R. H. (1969), "Numerically Stable Explicit Integration Techniques using a Linearized Runge-Kutta Extension," *Boeing Scientific Laboratories Document dl*-82-0929.

Anderson, N. H., Ball, R. B., and Voss, J. R. (1960), "A Numerical Method for Solving Control Differential Equations on Digital Computers," *JACM*, **7**, pp. 61–68.

Axelsson, O. (1969), "A Class of *A*-Stable Methods," *BIT*, **9**, pp. 185–199.

Bard, A., Ceschino, F., Kuntzmann, J., and Laurent, P. (1961), "Formules de base de la méthode de Runge-Kutta," *Chiffres*, **4**, pp. 31–37.

Bashforth, F. and Adams, J.C. (1883), *Theories of Capillary Action.*, Cambridge U. P., New York.

Birkhoff, G. and Varga, R. S. (1965), "Discretization Errors for Well Set Cauchy Problems," *Jour. Math. and Phys.*, **44**, pp. 1–23.

Bjurel, G. (1969), "Preliminary Report on Modified Linear Multistep Methods for a Class of Stiff Ordinary Differential Equations," *Dept. of Information Processing*, The Royal Institute of Technology, Stockholm, Report # NA 69.02.

Bjurel, G., Dahlquist, G., Lindberg, B., Linde, S., and Odén, L. (1970), "Survey of

Stiff Ordinary Differential Equations, *Dept. of Information Processing*, Royal Institute of Technology, Stockholm, Report # NA 70.11.

Blum, E. K. (1962), "A Modification of the Runge-Kutta Fourth Order Method," *Math. Comp.*, **16**, pp. 176–187.

Brayton, R. K., Gustavson, F. G., and Liniger, W. (1966), "A Numerical Analysis of the Transient Behavior of a Transistor Circuit,", *IBM Jour.* **10**, pp. 292–299.

Brock, P. and Murray, F. J. (1952), "The Use of Exponential Sums in Step by Step Integration," *MTAC*, **6**, pp. 63–78, 138–150.

Brown, R. R., Riley, J. D. and Bennett, M. M. (1965), "Stability Properties of Adams-Moulton Type Methods," *Math. Comp.*, **19**, pp. 90–96.

Brush, D. G., Kohfeld, J. J. and Thomson, G. T. (1967), "Solution of Ordinary Differential Equations Using Two Off-Step Points," *JACM*, **14**, pp. 769–784.

Bulirsch, R. and Stoer, J. (1964), "Fehlerabschatzungen und Extrapolation mit rationalen Funktioner bei Verfahren von Richardson-typus," *Num. Math.*, **6**, pp. 413–427.

Bulirsch, R. and Stoer, J. (1966), "Numerical Treatment of Ordinary Differential Equations by Extrapolation Methods," *Num. Math.*, **8**, pp. 1–13.

Bulirsch, R. and Stoer, J. (1966B), "Asymptotic Upper and Lower Bounds for Results of Extrapolation Methods," *Num. Math.*, **8**, pp. 93–104.

Butcher, J. C. (1963), "Coefficients for the Study of Runge-Kutta Integration Processes," *Jour. Australian Math. Society*, **3**, pp. 185–201.

Butcher, J. C. (1964), "Implicit Runge-Kutta Processes,", *Math, Comp,* **18**, pp. 50–64.

Butcher, J. C. (1964B), "On Runge-Kutta Processes of High Order," *Jour. Australian Math. Society*, **4**, pp. 179–194.

Butcher. J. C. (1964C), "Integration Processes Based on Radau Quadrature Formulas," *Math. Comp.*, **18**, pp. 233–244.

Butcher, J. C. (1965), "On the Attainable Order of Runge-Kutta Methods," *Math, Comp.*, **19**, pp. 408–417.

Butcher, J. C. (1965B), "A Modified Multistep Method for the Numerical Integration of Ordinary Differential Equations," *JACM*, **12**, pp. 124–135.

Butcher, J. C. (1966), "On the Convergence of Numerical Solutions to Ordinary Differential Equations," *Math. Comp.*, **20**, pp. 1–10.

Butcher, J. G. (1967), "A Multistep Generalization of Runge-Kutta with Four or Five Stages," *JACM*, **14**, pp. 84–99.

Byrne, G. D. (1967) "Paramameters for Pseudo Runge-Kutta Methods," *Comm ACM* **10**, No. 2, pp. 102–104

Byrne, G. D. and Lambert, R. J. (1966), "Pseudo Runge-Kutta Methods Involving Two Points," *JACM*, **13**, pp. 114–123.

Calahan D. (1969), "Numerical Considerations in the Transient Analysis and Optimal Design of Nonlinear Circuits," *Digest Record of Joint Conference on Mathematical and Computer Aids to Design*, ACM/SIAM/IEEE, Anaheim, Cal., pp. 129–145.

Calahan, D. A. (1967), "Numerical Solution of Linear Systems with Widely Separated Time Constants," *Proc. IEEE*, **55**, pp. 2016–2017.

Calahan, D. A. (1968), "A Stable Accurate Method of Numerical Integration for Non-Linear Systems," *Proc. IEEE*, **56**, p. 744.

Calahan, D. A. and Abbott, N. E. (1970), "Stability Analysis of Numerical Integration," *Proc. of the Tenth Midwest Symposium on Circuit Theory*, pp. I-2-1 to I-2-20.

Calahan, D. A. and Gear, C. W. (1969), "An Ill-Conditioning Problem with Implicit Integration," *Proc. IEEE*, **57**, pp. 1775–1776.

Case, J. (1969), "A Note on the Stability of Predictor Corrector Techniques," *Math. Comp.*, **23**, pp. 741–750.

Casity, C. R. (1966), "Solutions of the Fifth Order Runge-Kutta Equations," *SINUM*, **3**, pp. 598–606.

Cassity, C. R. (1969), "The Complete Solution of the Fifth Order Runge-Kutta Equations," *SINUM*, **6**, No. 3, pp. 432–436.

Certaine, J. (1960), "The Solution of Ordinary Differential Equations with Large Time Constants," in *Mathematical Methods for Digital Computers*, ed. A. Ralston and H. S. Wilf. Wiley, New York, pp. 128–132.

Ceschino, F. (1961), "Modification de la longueur du pas dan l'intégration numérique par les méthods à pas liés," *Chiffres*, **2**, pp. 101–106.

Ceschino, F. (1961B), "Une méthode de mise en oeuvre des formules d'Obrechkoff pour l'intégration des équations différentielles," *Chiffres*, **2**, pp. 49–54.

Ceschino, F. and Kuntzmann, J. (1963), *Problèmes différentiels de conditions initials*. Dunod, Paris. Translation by D. Boyanovitch, as *Numerical Solution of Initial Value Problems*. Prentice-Hall, Englewood Cliffs, N. J., 1966.

Chase, P. E. (1962), "Stability Properties of Predictor-Corrector Methods for Ordinary Differential Equations," *JACM*, **9**, pp. 457–468.

Christiansen, J. (1970), "Handbook Series Numerical Integration, Numerical Solution of Ordinary Simultaneous Differential Equations of the First Order Using a Method for Automatic Step Change," *Num. Math*, **14**, No. 4, pp. 317–324.

Clark, N. A. (1966), "Program Description for Library Subroutine ANL D250 DIFSUB," Argonne National Laboratory, Argonne, Ill.

Clark, N. A. (1968), "A Study of Some Numerical Methods for the Integration of Systems of First Order Ordinary Differential Equations," *Argonne National Lab Report No.* 7428.

Clenshaw, C. W. (1957), "The Numerical Solution of Linear Differential Equations in Chebyshev Series," Proc. Cambridge Phil. Society, **53**, pp. 134–149.

Clenshaw, C. W. (1960), "The Numerical Solution of Ordinary Differential Equations in Chebyshev Series," in *PICC Symposium, Rome, on Differential and Integral Equations*. Birkhäuser, Basel, pp. 222–227.

Clenshaw, C. W. and Curtiss, A. R. (1960), "A Method for Numerical Integration on an Automatic Computer," *Num. Math.*, **2**, pp. 197–205.

Cohen, C. J. and Hubbard, E. C. (1960), "An Algorithm Applicable to Numerical Integration of Orbits in Multiple Revolution Steps," *Astron. J.*, **65**, pp. 454–456.

Collatz, L. (1960), *The Numerical Treatment of Differential Equations*, 3rd ed. Springer, Berlin.

Cooper, G. J. (1967), "A Class of Single Step Methods for Systems of Nonlinear Differential Equations," *Math. Comp.*, **21**, pp. 597–610.

Cooper, G. J. (1968), "Interpolation and Quadrature Methods for Ordinary Differential Equations," *Math. Comp.*, **22**, 69–76.

Cooper, G. J. and Gal, E. (1967), "Single Step Methods for Linear Differential Equations," *Num. Math.*, **10**, pp. 307–315.

Courant, R. (1936), *Differential and Integral Calculus*, Vol. 2. Interscience, New York.

Cowell, P. H. and Crommelin, A. C. D (1910), "Investigation of the Motion of Halley's Comet from 1759 to 1910," appendix to *Greenwich Observations for 1909.* Edinburgh. p. 84.

Crane, R. L. and Klopfenstein, R. W. (1965), "A Predictor-Corrector Algorithm with Increased Range of Absolute Stability," *JACM*, pp. 227–241.

Crane, R. L. and Lambert, R. J. (1962), "Stability of a Generalized Corrector Formula," *JACM*, **9**, No. 1. pp. 104–117.

Curtiss, C. F. and Hirschfelder, J. O. (1952), "Integration of Stiff Equations," *Proc. Nat. Acad. Science, U. S.*, **38**, pp. 235–243.

Dahlquist, G. (1956), "Numerical Integration of Ordinary Differential Equations," *Math. Scandinavica*, **4**, pp. 33–50.

Dahlquist, G. (1959), "Stability and Error Bounds in the Numerical Integration of Ordinary Differential Equations," *Trans. Roy. Inst. Tech., Stockholm*, No. 130.

Dahlquist, G. (1963), "A Special Stability Problem for Linear Multistep Methods," *BIT*, **3**, pp. 27–43.

Dahlquist, G. (1963B), "Stability Questions for Some Numerical Methods for Ordinary Differential Equations," *Proc. Symposium for Applied Math.*, **15**, pp. 147–158.

Dahlquist, G. (1966), "On Rigorous Error Bounds in the Numerical Solution of Ordinary Differential Equations," in *The Numerical Solution of Nonlinear Differential Equations*, ed. D. Greenspan. John Wiley and Sons, New York, pp. 89–96.

Dahlquist, G. (1969), "A Numerical Method for some Ordinary Differential Equations with Large Lipshitz Constants," in *Information Processing 68*, ed. A. J. H. Morrell. North Holland Publishing Co., Amsterdam, pp. 183–186.

Danchick, R. (1968), "Further Results on Generalized Predictor-Corrector Methods," *Jour. Comp. and Sys. Sciences*, **2**, No. 2, pp. 203–218.

Davison, E. (1967), "A High Order Crank-Nicholson Technique for Solving Differential Equations," *Comp. J.*, **10**, pp. 195–197.

Day, J. T. (1964), "A One-Step Method for the Numerical Solution of Second Order Linear Ordinary Differential Equations," *Math. Comp.*, **18**, p. 664.

Day, J. T. (1965), "A Runge-Kutta Method for the Numerical Integration of the Differential Equation $y'' = f(x, y)$," *ZAMM*, **5**, pp. 354–356.

Day, J. T. (1965B), "A One-Step Method for the Numerical Integration of the Differential Equation $y'' = f(x)y + g(x)$," *Comp. J.*, **7**, p. 314.

Decell, H. P., Jr., Guseman, L. F. and Lea, R. N. (1966), "Concerning the Numerical Solution of Differential Equations," *Math. Comp.*, **20**, No. 95, pp. 431–434.

DeGroat, J. J. and Abbett, M. J. (1965), "A Computation of One-Dimensional Combustion of Methane," AIAA Jour., **3**, pp. 381–383.

Dejon, B. (1966), "Stronger than Uniform Convergence of Multistep Difference Methods," *Num. Math.*, **8**, pp. 29–41.

Dejon, B. (1967), "Numerical Stability of Difference Equations with Matrix Coefficients," *SINUM*, **4**, No. 1, pp. 119–128.

Dennis, S. C. R. (1960), "The Numerical Integration of Ordinary Differential Equations Possessing Exponential Type Solutions." *Proc. Cambridge Phil. Society*, **56**, pp. 240–246.

Dennis, S. C. R. (1962), "Step by Step Integration of Ordinary Differential Equations," *Applied Math. Quarterly*, **20**, pp. 359–372.

Descloux, J. (1963), "Note on a Paper by A. Nordsieck," *Department of Computer Science Report No.* 131, University of Illinois, Urbana, Ill.

Dill, C. (1969), "A Computer Graphic Technique for Finding Numerical Methods for Ordinary Differential Equations," *Department of Computer Science Report No.* 295, University of Illinois, Urbana, Ill.

Dyer, J. (1968), "Generalized Multistep Methods in Satellite Orbit Computation," *JACM*, **15**, No. 4, pp. 712–719.

Ehle, B. L. (1968), "High Order *A*-stable Methods for the Numerical Solution of Systems of Differential Equations," *BIT*, **8**, pp. 276–278.

Emanuel, G. (1964), "Numerical Analysis of Stiff Equations," *Aerospace Report No.* TDR-269 (4230-20)-3.

Engeli, M. E. (1969), "Achievements and Problems in Formula Manipulation," in *Information Processing, 68*, ed. A. J. H. Morrell. North Holland Publishing Co., Amsterdam: pp. 24-32.

Fehlberg, E. (1960), "Neue genauere Runge-Kutta Formeln fur Differentialgleichungen *n*-ter Ordung," *ZAMM*, **40**, pp. 449–455.

Fehlberg, E. (1966), "New High Order Runge-Kutta Formulas with an Arbitrary Small Truncation Error," ZAMM, **46**, pp. 1–16.

Feldstein, M. A. and Stetter, H. J. (1963), "Simplified Predictor-Corrector Methods," *Proceedings 18th ACM National Conference.*

Forrington, C. V. D. (1961), "Extensions of the predictor-corrector method for the solution of systems of ordinary differential equations," *Comp. J.*, **4**, pp. 80–84.

Forsythe, G. and Moler, C. B. (1967), *Computer Solution of Linear Algebraic Systems.* Prentice-Hall, Inc., Englewood Cliffs, N. J.

Fowler, M. E. and Warten, R. M. (1967), "A Numerical Integration Technique for Ordinary Differential Equations with Widely Separated Eigenvalues," IBM Jour., **11**, pp. 537–543.

Fox, L. (1962), "Chebyshev Methods for Ordinary Differential Equations," *Comp. J.*, **4**, pp. 318–331.

Fox, L. (1962), *Numerical Solution of Ordinary and Partial Differential Equations.* Pergamon, New York.

Froese, C. (1961), "An evaluation of Runge-Kutta Type Methods for Higher Order Differential Equations," *JACM*, **8**, pp. 637–644.

Fyfe, D. J. (1966), "Economical Evaluation of Runge-Kutta Formulas," *Math. Comp.*, **20**, No. 95, pp. 392–398.

Gabel, G. (1968), "Predictor Corrector Methods using Divided Differences," Master's thesis, University of Toronto.

Gates, L. D., Jr. (1964), "Numerical Solution of Differential Equations by Repeated Quadratures," *SIAM Review*, **6**, pp. 134–147.

Gautschi, W. (1961), "Numerical Integration of Ordinary Differential Equations Based on Trigonometric Polynomials," *Num. Math.*, **3**, pp. 381–397.

Gear, C. W. (1965), "Hybrid Methods for Initial Value Problems in Ordinary Differential Equations," *SINUM*, **2**, p. 69.

Gear, C. W. (1966), "The Numerical Integration of Ordinary Differential Equations of Various Orders," *Argonne National Lab Report*, #ANL 7126.

Gear, C. W. (1967), "The Numerical Integration of Ordinary Differential Equations," *Math. Comp.*, **21**, pp. 146–156.

Gear, C. W. (1969), "The Automatic Integration of Stiff Ordinary Differential Equations," in *Information Processing, 68*, ed. A. J. H. Morrel. North Holland Publishing Company, Amsterdam. pp. 187–193.

Gear, C. W. (1970), "Rational Approximations by Implicit Runge-Kutta Schemes, *BIT*, **10**, pp. 20–22.

Gear C. W. (1971) "The Simulataneous Numerical Solution of Differential-Algebraic Systems," *IEEE Transactions on Circuit Theory*, **18**, No. 1, pp. 89–95.

Giese, C. (1967), "State Variable Difference Methods for Digital Simulation, *IEEE Transactions on Computers*, **8**, pp. 263–271.

Giloi, W. and Grebe, H (1968), "Construction of Multistep Integration Formulas for Simulation Purposes," *IEEE*, **17**, No. 12, pp. 1121–1131.

Gragg, W. (1963), "Repeated Extrapolation to the Limit in the Numerical Solution of Ordinary Differential Equations," Doctoral dissertation, UCLA.

Gragg, W. (1965), "On Extrapolation Algorithms for Ordinary Initial Value Problems," *SINUM*, **2**, pp. 384–403.

Gragg, W. and Stetter, H. (1964), "Generalized Multistep Predictor-Corrector Methods," *JACM*, **11**, No. 2, pp. 188–209.

Greenspan, H., Hafner, W., and Ribaric, M. (1965), "On Varying Step Sizes in Numerical Integration of First Order Differential Equations," *Num. Math.*, **7**, pp. 286–291.

Hachtel, G., Brayton, R., and Gustavson, F. (1971), "The Sparse Tableux Approach to Network Analysis and Design," *IEEE Transactions on Circuit Theory*, **18**, No. 1.

Hafner, P. A. (1969), "Stability Charts of Various Numerical Methods for Solving Systems of Ordinary Differential Equations," *Weapons Research Establishment Technical Note* WSD 112, Salisbury, South Australia, November, 1969

Hain, K. and Hertweck, F. (1960), "Numerical Integration of Ordinary Differential Equations by Difference Methods with Automatic Determination of Steplength," in *PICC Symposium, Rome, Differential and Integral Equations.* Birkhauser, Basel.

Haines, C. F. (1969), "Implicit Integration Processes with Error Estimate for the Numerical Solution of Differential Equations," *Comp. J.*, **12**, No. 2, pp. 183–187.

Hansen, K. F., Koen, B. V., and Little, W. W. (1966), "Stable Numerical Solutions of the Reactor Kinetics Equations," *Nuc. Science Eng.*, **25**, pp. 183–188.

Henrici, P. (1962), *Discrete Variable Methods for Ordinary Differential Equations.* John Wiley and Sons, New York.

Henrici, P. (1963), *Error Propagation for Difference Methods.* John Wiley and Sons, New York.

Henrici, P. (1964), *Elements of Numerical Analysis.* Wiley, New York.

Heun, K. (1900), "Neue Methode zur approximativen Integration der Differentialgleichungen einer unabhängigen Veränderlichen," *Z. Math. u. Phys.*, **45**, pp. 23–38.

Hildebrand, F. B. (1956), *Introduction to Numerical Analysis.* McGraw-Hill Book Co., New York.

Hildbrand, F. B. (1968), *Finite Difference Equations and Simulation.* Prentice-Hall, Inc., Englewood Cliffs, N. J.

Hull, T. E. (1962), "Corrector Formulas for Multistep Integration Methods," *SIAM Jour.*, **10**, pp. 351–369.

Hull, T. E. (1967), "A Search for Optimum Methods for the Numerical Integration of Ordinary Differential Equations," *SIAM Review*, **9**, 647–654.

Hull, T. E. (1968), "The Effectiveness of Numerical Methods for Ordinary Differential Equations," *Studies in Numerical Analysis*, **2**, pp. 114–121.

Hull, T. E. (1969), "The Numerical Integration of Ordinary Differential Equations," in *Information Processing 68*, ed. A. J. H. Morrell. North Holland Publishing Company, Amsterdam.

Hull, T. E. and Creemer, A. L. (1963), "Efficiency of Predictor-Corrector Schemes," *JACM*, **10**, pp. 291–301.

Hull, T. E. and Johnston, R. L. (1964), "Optimum Runge-Kutta Methods," *Math. Comp.*, **18**, pp. 306–310.

Hull, T. E. and Swenson, J. R. (1966), "Tests of Probabilistic Models for Propogation of Round-Off Error,", CACM, **9**, pp. 108–113.

Imhof, J. P. (1963), "On the Method for Numerical Integration of Clenshaw and Curtis," *Num. Math.*, **5**, pp. 138–141.

Ince, E. L. (1956), *Ordinary Differential Equations.* Dover, New York.

Ira, M. (1964), "A Stabilizing Device for Unstable Solutions of Ordinary Differential Equations, Design and Application of a Filter," *Information Processing in Japan*, **4**, pp. 65–73.

Isaacson, E. and Keller H. B. (1966), *Analysis of Numerical Methods.* John Wiley and Sons, Inc., New York.

Jain, M. K. and Srivastava, V. K. (1970), "High Order Stiffly Stable Methods for Ordinary Differential Equations," *Department of Computer Science Report No.* 394, University of Illinois, Urbana, Ill.

Kahan, W. (1966), "A Computable Error Bound for Systems of Ordinary Differential Equations," *SIAM Review* (*Abstract*), **8**, pp. 568–569.

Kaluza, T. (1928), "Uber die Koefficienten reziproker Petenzreihen," *Math. Zs.*, **28**, pp. 161–170.

Karim, A. I. A. (1966), "The Stability of the Fourth Order Runge-Kutta Method for the Solution of Systems of Differential Equations," *CACM*, **9**, pp. 113–116.

Karim, A. I. A. (1968), "A Theorem for the Stability of General Methods for the Solution of Differential Equations," *JACM*, **15**, No. 4, pp. 706–711.

Keller, H. B. (1968), *Numerical Methods for Two-point Boundary Value Problems.* Blaisdell, Waltham, Mass.

King, R. (1966), "Runge-Kutta Methods with Constrained Minimum Error Bounds," *Math. Comp.*, **20**, No. 95, pp. 386–391.

Klopfenstein, R. W. and Millman, R. S. (1968), "Numerical Stability of One-Evaluation Predictor-Corrector Methods," *Math. Comp.*, **22**, No. 103, pp. 557–564.

Kohfeld, J. J. and Thompson, G. T. (1967), "Multistep Methods with Modified Predictors and Correctors," *JACM*, **14**, pp. 155–166.

Kohfeld, J. J. and Thompson, G. T. (1968), "A Modification of Nordsieck's Method using an Off Step Point," *JACM*, **15**, No. 3, pp. 390–401.

Konen, H. P. and Luther, H. A. (1967), "Some Singular Explicit Fifth Order Runge-Kutta Solutions," *SINUM*, **4**, pp. 607–619.

Kopal, Z. (1961), *Numerical Analysis*, 2nd ed. Wiley, New York.

Kowalik J. and Osborne M. R. (1968), *Methods for Unconstrained Optimization Problems.* American Elsevier Co., New York.

Krogh, F. T. (1966), 'Predictor-Corrector Methods of High Order with Improved Stability Characteristics," *JACM*, **13**, pp. 374–385.

Krogh, F. T. (1967), "A Note on the Effect of Conditionally Stable Correctors," *Math. Comp.*, **21**, No. 100, pp. 717–719.

Krogh, F. T. (1967B), "A Test for Instability in the Numerical Solution of Ordinary Differential Equations," *JACM*, **14**, pp. 351–354.

Krogh, F. T. (1967C), "On Methods of Adams' Type for the Numerical Solution of Ordinary Differential Equations," *TRW Report* No. 67.3122.2.317.

Krogh, F. T. (1969), "A Variable Step, Variable Order Multistep Method for the Numerical Solution of Ordinary Differential Equations," in *Information Processing 68*, Vol. I, ed. A. J. H. Morrell. North Holland Publishing Company, Amsterdam, pp. 194–199.

Krogh, F. T. (1969B), "On Testing a Subroutine for the Numerical Integration of Ordinary Differential Equations," *Jet Propulsion Lab Tech. Report* #217.

Kruckberger, F. and Unger, H. (1960), "On the Numerical Integration of Ordinary Differential Equations and the Determination of Error Bounds," in *PICC*

Symposium, Rome, on Differential and Integral Equations. Birkhauser, Basel. pp. 369–379.

Kuntzmann, J. (1961), "Neuere Entwickelungen der Methode von Runge und Kutta," *ZAMM*, **41**, pp. 28–31.

Kuntzmann, J. (1962), "Nouvelle méthode pour l'intégration approchée des équations différentielles," in *Information Processing 62*, ed. C. Popplewell, North Holland Publishing Company, Amsterdam.

Lambert, R. J. (1967), "An Analysis of the Numerical Stability of Predictor-Corrector Solutions of Nonlinear Ordinary Differential Equations," *SINUM*, **4**, No. 4, pp. 597–606.

Lambert, J. and Mitchell, A. (1962), "On the Solution of $y' = f(x, y)$ by a Class of High Accuracy Difference Formulas of Low Order," *Z. Angew. Math. Phys.*, **13**, pp. 223–232.

Lambert, J. and Shaw, B. (1965), "Numerical Solution of $y' = f(x, y)$ by a Class of Formulas Based on Rational Approximation," *Math. Comp.*, **19**, No. 91, pp. 456–462.

Lambert, J. D. and Shaw, B. (1966), "A Generalization of Multistep Methods for Ordinary Differential Equations," *Num. Math.*, **8**, pp. 250–263.

Lambert, J. D. and Shaw, B. (1966B), "A Method for the Numerical Solution of $y' = f(x, y)$ Based on a Self-Adjusting Nonpolynomial Interpolant," *Math. Comp.*, **20**, pp. 11–20.

Lanczos, C. (1960), "Solution of Ordinary Differential Equations by Trigonometric Interpolation," in *PICC Symposium, Rome, on Differential and Integral Equations.* Birkhäuser, Basel. pp. 22–32.

Laurent, P. J. (1961), "Méthodes spéciales du type de Runge-Kutta," *Premier Congress AFCAL*, pp. 27–36.

Lawson, J. D. (1966), "An Order Five Runge-Kutta Process with Extended Region of Stability," *SINUM*, **3**, pp. 593–597.

Lawson J. D. (1967), "Generalized Runge-Kutta Processes for Stable Systems with with Large Lipschitz Constants," *SINUM*, **4**, pp. 372–380.

Lawson, J. D. (1967B), "An Order Six Runge-Kutta Process with Extended Region of Stability," *SINUM*, **4**, pp. 620–625.

Lee, H. B. (1967), "Matrix Filtering as an Aid to Numerical Integration," *Proc. IEEE*, **55**, pp. 1826–1831.

Lether, F. G. (1966), "The Use of Richardson Extrapolation in One-Step Methods with Variable Step Size," *Math. Comp.*, **20**, No. 95, pp. 379–385.

Lewis, H. R. and Stovall, E. J., Jr. (1967), "Comments on a Floating-Point Version of Nordsieck's Scheme for the Numerical Integration of Differential Equations," *Math. Comp.*, **21**, pp. 157–161.

Liniger, W. (1968), "Optimization of a Numerical Method for Stiff Systems of Ordinary Differential Equations," *IBM Research Report No.* RC-2198.

Liniger, W. (1968), "A Criteria for A-Stability of Linear Multistep Integration Formulae," *Computing*, **3**, pp. 280–285.

Liniger, W. (1969), "Global Accuracy and *A*-Stability of One- and Two-Step Integration Formulae for Stiff Ordinary Differential Equations," *IBM Research Report No.* RC-2396.

Liniger, W. and Willoughby, R. (1970), "Efficient Integration Method for Stiff Systems of Ordinary Differential Equations," *SINUM*, **7**, No. 1, pp. 47–66.

Liou, M. L. (1966), "A Novel Method of Evaluating Transient Response," *Proc. IEEE*, **54**, No. 1, pp. 20–23.

Lomax, H. (1968), "On the Construction of Highly Stable Explicit Numerical Methods for Integration of Coupled Ordinary Differential Equations with Parasitic Eigenvalues," *NASA Tech. Report No. TN* 4547.

Lomax, H. (1968B), "Stable Implicit and Explicit Numerical Methods for Integrating Quasi-Linear Differential Equations with Parasitic-Stiff and Parasitic-Saddle Eigenvalues," *NASA Tech. Note NASA, No. TN* D-4703 Ames Research Center, Moffett Field, Calif.

Lomax, H. and Bailey, H. E. (1967), "A Critical Analysis of Various Numerical Integration Methods for Computing the Flow of a Gas in Chemical Non-Equilibrium," *NASA Tech. Report No. TN* D-4109.

Loscalzo, F. R. (1969), "An Introduction to the use of Spline Functions in Ordinary Differential Equations," in *Theory and Applications of Spline Functions*, ed. T. N. E. Greville. Academic Press, New York, pp. 37–64.

Loscalzo, F. R. and Talbot, T. D. (1967), "Spline Function Approximations for Solution of Ordinary Differential Equations," *SINUM*, **4**, No. 3, pp. 433–445.

Lotkin, M. (1951), "On the Accuracy of Runge-Kutta's Method," *MTAC*, **5**, pp. 128–132.

Luther, H. A. (1966), "Further Explicit Fifth Order Runge-Kutta Formulas," *SIAM Review*, **8**, pp. 374–380.

Luther, H. A (1968), "An Explicit Sixth Order Runge-Kutta Formula," *Math. Comp*., **22**, No. 102, pp. 344–346.

Luther, H. A. and Konen, H. P. (1965), "Some Fifth Order Classical Runge-Kutta Formulas," *SIAM Review*, **7**, pp. 551–558.

Magnus, D. and Schecter, H., "Analysis and Application of the Pade Approximation for the Integration of Chemical Kinetic Equations," *General Applied Science Labs Tech. Report No.* 642, (Project 5810).

Makinson, G. J. (1968), "Stable High Order Implicit Methods for the Numerical Solution of Systems of Differential Equations," *Comp. J.*, **11**, pp. 305–310.

Merson, R. H. (1957), "An Operational Method for the study of Integration Processes," *Proceedings of a Symposium on Data Processing*, Weapons Research Establishment, Salisbury, South Australia. A description of the Runge-Kutta-Merson method can be found in Fox (1962).

Miller, J. C. P. (1966), "The Numerical Solution of Ordinary Differential Equations," in *Numerical Analysis: An Introduction*, ed. J. Walsh. Academic Press, London. pp. 63–98.

Milne, W. E. and Reynolds, R. R. (1962), "Fifth-Order Methods for the Numerical Solution of Ordinary Differential Equations," *JACM*, **9**, pp. 64–70.

Miranker, W. L. (1968), "Difference Schemes for the Integration of Stiff Systems of Ordinary Differential Equations," *IBM Research Report No.* RC-1977.

Miranker, W. L. and Liniger, W. (1967), "Parallel Methods for the Numerical Integration of Ordinary Differential Equations," *Math. Comp.*, **21**, pp. 303–320.

Moore, R. E. (1965), "Automatic Local Coordinate Transformations to Reduce the Growth of Error Bounds in the Interval Computation of Solutions of Ordinary Differential Equations," in *Error in Digital Computation*, Vol. 2, ed. L. B. Ball. Wiley, New York.

Moore, R. E. (1966), *Interval Analysis*. Prentice-Hall, Englewood Cliffs, N. J,

Moretti, G. (1965), "A New Technique for the Numerical Analysis of Nonequilibrium Flows," *AIAA Jour.*, **3**, pp. 381–383.

Morrison, D. (1962), "Optimal Mesh Size in the Numerical Integration of an Ordinary Differential Equation," *JACM*, **9**, pp. 98–103.

Moulton, F. R. (1926), *New Methods in Exterior Ballistics*. U. of Chicago, Chicago.

Mysovskikh, I. P. (1969), *Lectures on Numerical Methods*. Translation by L. B. Ball. Wolters-Noordhoff Publ. Co., Groningen, Netherlands.

Newberry, A. C. R. (1963), "Multistep Integration Formulas," *Math. Comp*, **17**, pp. 452–455.

Newberry, A. C. R. (1967), "Convergence of Successive Substitution Starting Procedures, *Math. Comp.*, **21**, No. 99, pp. 489–491.

Nordsieck, A. (1962), "On the Numerical Integration of Ordinary Differential Equations," *Math. Comp.*, **16**, pp. 22–49. .

Norsett, S. P. (1969), "A Criterion for $A(\alpha)$ Stability of Linear Multistep Methods," *BIT*, **9**, pp. 259–263.

Nugeyre, J. B. (1961), "Un procédé mixte (Runge-Kutta, pas liés) d'intégration des systèmes différentiels du type $x'' = X(x, t)$," *Chiffres*, **4**, pp. 55–68.

Osborne, M. R. (1964), "A Method for Finite Difference Approximation to Ordinary Differential Equations," *Comp. J.*, **7**, pp. 58–65.

Osborne, M. R. (1967), "Minimizing Truncation Error in Finite Difference Approximations to Ordinary Differential Equations," *Math. Comp.*, **21**, No. 98, pp. 133–145.

Osborne, M. R. (1969), "A New Method for the Integration of Stiff Systems of Ordinary Differential Equations," in *Information Processing 68*, ed. A. J. H. Morrel. North Holland Publishing Company, Amsterdam, pp. 200–204.

Pope, D. A. (1963), "An Exponential Method of Numerical Integration of Ordinary Differential Equations." *Comm. ACM*, **6**, pp. 491–493.

Rahme, H. S. (1969), "A New Look at the Numerical Integration of Ordinary Differential Equations," *JACM*, **16**, No. 3, pp. 496–506.

Ralston, A. (1961), "Some Theoretical and Computational Matters Relating to Predictor Corrector Methods of Numerical Integration," *Comp. J.*, **4**, pp. 64–67.

Ralston, A. (1962), "Runge-Kutta with Minimum Error Bounds," *Math. Comp.*, **16**, pp. 431–437 (1962); *Math Comp.*, **17**, p. 488 (1963).

Ralston, A. (1965), *A First Course in Numerical Analysis*. McGraw-Hill, New York.

Ralston, A. (1965B), "Relative Stability in the Numerical Solution of Ordinary Differential Equations, *SIAM Review*, **7**, pp. 114–125.

Reimer, M. (1968), "Finite Difference Forms Containing Derivatives of Higher Order," *SINUM*, **5**, No. 4, pp. 725–738.

Richards, P. I., Lanning, W. D., and Torrey, M. D. (1965), "Numerical Integration of Large, Highly-Damped Nonlinear Systems," *SIAM Review*, **7**, No. 3, pp. 376–380.

Richardson, L. F. (1927), "The Deferred Approach to the Limit, I—Single Lattice," *Trans. Roy. Soc., London*, **226**, pp. 299–349.

Robertson, H. H. (1966), "The Solution of a Set of Reaction Rate Equations," in *Numerical Analysis: An Introduction*, ed. J. Walsh. Academic Press, London, pp. 178–182.

Roe, G. M. (1967), "Experiments with a New Integration Algorithm," G. E. Report *No*. 67-C-037, Schenectady, N. Y.

Rosenbrock, H. H. (1963), "Some General Implicit Processes for the Numerical Solution of Differential Equations," *Comp*. J. **5** pp. 329–330.

Rosser, J. B. (1967), "A Runge-Kutta for all Seasons," *SIAM Review*, **9**, No. 3, pp. 417–452.

Rutishauser, H. (1960), "Bemerkungen zur numerischen Integration gewohnlicher Differentialgleichungen n-ter Ordnung," *Num. Math.*, **2**, pp. 263–279.

Sachnoff, L. (1960), "Integration of Simultaneous Differential Equations Using Multiple Stepsizes," *15th ACM National Conference.*

Sandberg, I. W. (1967), "Some Properties of a Class of Numerical Integration Formula," *Bell System Tech. Jour.*, **46**, No. 9, pp. 2061–2080.

Sandberg, I. W. (1967B), "Two Theorems on the Accuracy of Numerical Solutions of Systems of Ordinary Differential Equations," *Bell System Tech. Jour.* **46**, No. 6, pp. 1243–1266.

Sandberg, I. W. and Schichman, H. (1968), "Numerical Integration of Systems of Stiff Nonlinear Differential Equations," *Bell System Tech. Jour.*, **47**, No. 4, pp. 511–528.

Sarafyan, D. (1965), "Multistep Methods for the Numerical Solution of Ordinary Differential Equations Made Self-Starting," *Mathematics Research Center Report No*. 495.

Scraton, R. E. (1964), "The Numerical Solution of Second Order Differential Equations Not Containing the First Derivative Explicitly," *Comp. J.*, **6**, pp. 368–370.

Scraton, R. E. (1964B), "Estimation of the Truncation Error in Runge-Kutta and Allied Processes," *Comp. J.*, **7**, pp. 246–248.

Scraton, E. E. (1965), "The Solution of Linear Differential Equations in Chebyshev Series," *Comp. J.*, **8**, pp. 57–61.

Shampine, L. F. and Watts, H. A. (1969), "Block Implicit One-Step Methods," *Math. Comp.*, **23**, No. 108, pp. 731–740.

Shanks, E. B. (1966), "Solutions of Differential Equations by Evaluations of Functions," *Math. Comp.*, **20**, No. 93, pp. 21–38.

Shaw, B. (1967), "Modified Multistep Methods Based on Nonpolynomial Interpolants," *JACM*, **14**, pp. 143–154.

Shaw, B. (1967B), "Some Multistep Formulas for Special High Order Ordinary Differential Equations," *Num. Math.*, **9**, pp. 367–378.

Silverberg, M. (1968), "A New Method of Solving State Variable Equations Permitting Large Step Sizes," *Proc. IEEE*, **56**, No. 8, pp. 1343–1352.

Sloat, H. and Bickart, T. A. (1970), "An Implicit Formula for the Integration of Stiff Network Equations," *Proc. of the Third Hawaii Interational Conference on Systems Science.*

Spijker, M. (1966), "Convergence and Stability of Step by Step Methods for the Numerical Solution of Initial-Value Problems," *Num. Math.*, **8**, pp. 161–177.

Squier, D. P. (1969), "One-Step Methods for Ordinary Differential Equations," *Num. Math.*, **13**, pp. 176–179.

Stetter, H. J. (1965), "Stabilizing Predictors for Weakly Stable Correctors," *Math. Comp.*, **19**, pp. 84–89.

Stetter, H. J. (1965B), "A Study of Strong and Weak Stability in Discretization Algorithms," *SINUM*, **2**, pp. 265–280.

Stetter, H. J. (1965C), "Asymptotic Expansions for the Error of Discretization Algorithms for Nonlinear Functional Equations," *Num. Math.*, **7**, pp. 18–31.

Stetter, H. J. (1968), "Improved Absolute Stability of Predictor-Corrector Schemes," *Computing*, **3**, pp. 286–296.

Stineman, R. W. (1965), "Digital Time-Domain Analysis of Systems with Widely Separated Poles," *JACM*, **12**, No. 2, pp. 286–293.

Stoer, J. (1961), "Uberzwei Algorithmen zur Interpolation mit Rationalen Funktionen," *Num. Math.*, **3**, pp. 285–304.

Störmer, C. (1907), "Sur les trajectoires des corpuscules électrisés" *Arch. Sci. Phys. Nat., Genève*, **24**, pp. 5–18, 113–158, 221–247.

Störmer, C. (1921), "Méthodes d'intégration numérique des équations différentielles ordinaires," *C. R. Congr. Intern. Math., Strasbourg*, pp. 243–257.

Tewarson, R. P. (1969), "Projection Methods for Solving Sparse Linear Equations," *Comp. J.*, **12**, No. 1, pp. 77–80.

Tewarson, R. P. (1967), "Solution of a System of Simultaneous Linear Equations with a Sparse Coefficient Matrix by Elimination Methods," *BIT*, **7**, pp. 226–239.

Tinney, W. F. and Walker, J. W. (1967), "Direct Solutions of Sparse Network Equations by Optimally Ordered Triangular Factorization," *Proc. IEEE*, **55**, No. 11, pp. 1801–1809.

Todd, J. (1962), ed. *A Survey of Numerical Analysis*. McGraw-Hill, New York.

Treanor, C. E. (1966), "A Method for the Numerical Integration of Coupled First Order Differential Equations with Greatly Different Time Constants," *Math. Comp.*, **20**, No. 93, pp. 39–45.

Tyson, T. J. (1964), "An Implicit Integration Method for Chemical Kinetics," *TRW Report* No. 9840-6002-RU000, Redondo Beach, Calif.

Van Wyk, R. (1970), "Variable Mesh Multistep Methods for Ordinary Differential Equations," *Jour. Comp. Physics*, **5** pp. 244–264.

Verner, J. H. (1969), "The Order of Some Implicit Runge-Kutta Methods," *Num. Math.*, **13**, pp. 14–23.

Watt, J. M. (1967), "The Asymptotic Discretization Error of a Class of Methods for Solving Ordinary Differential Equations," Proc. Cambridge Phil. Society, pp. 441–472.

Whitney, D. E. (1966), "Propagated Error Bounds for Numerical Solution of Transient Response," Proc. *IEEE*, **54**, No. 8, pp. 1084–1085.

Whitney, D. E., (1969), "More about Similarities Between Runge-Kutta and Matrix Exponential Methods for Evaluating Transient Response," Proc. *IEEE*, **57**, No. 11, pp. 2053–2054.

Widlund, O. (1967), "A Note on Unconditionally Stable Linear Multistep Methods," *BIT*, **7**, pp. 65–70.

Willoughby, R. A. (1969), ed. "Proceedings of the Symposium on Sparse Matrixes and their Applications," *Report RA* 1 (#11707), IBM Watson Research Center, Yorktown Heights, N. Y.

Zajac, E. E. (1964), "Note on Overly-Stable Difference Approximation," *Jour. Math. and Phys.*, **18**, No. 1, pp. 51–54.

Zurmuhl, R. (1948), "Runge-Kutta-Verfahren zur numerischen Integration von Differentialgleichungen *n*-ter Ordnung," *ZAMM*, **28**, pp. 173–182.